

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Manual de código de la aplicación QUIZU

Autor: Francisco José Simoni García

Director: Dr. Rafael Muñoz Salinas

Agosto, 2021



UNIVERSIDAD DE CÓRDOBA

Contenido

Contenido.....	1
1 Introducción.....	3
2 Ficheros de funcionamiento.....	4
2.1 AuthActivity.kt.....	4
2.2 Registro.kt.....	7
2.3 HombreActivity.kt.....	10
2.4 HomeAlumno.kt.....	13
2.5 AsignaturaAlumno.kt.....	17
2.6 AsignaturaProfesor.kt.....	19
2.7 editarPerfil.kt.....	22
2.8 FormAsignatura.kt.....	23
2.9 FormQuestion.kt.....	25
2.10 FormQuiz.kt.....	32
2.11 Lista.kt.....	37
2.12 Seleccionado.kt.....	56
2.13 SolveQuiz.kt.....	68
2.14 Notas.kt.....	80
3 Ficheros de interfaz.....	92
3.1 activity_asignatura_alumno.xml.....	92
3.2 activity_asignatura_profesor.xml.....	94
3.3 activity_auth.xml.....	97
3.4 activity_editar_perfil.xml.....	98
3.5 activity_form_asignatura.xml.....	100
3.6 activity_form_question.xml.....	102
3.7 activity_form_quiz.xml.....	105
3.8 activity_home.xml.....	108
3.9 activity_home_alumno.xml.....	110
3.10 activity_lista.xml.....	112
3.11 activity_notas.xml.....	114
3.12 activity_registro.xml.....	115
3.13 activity_seleccionado.xml.....	118
3.14 activity_solve_quiz.xml.....	120
4 Ficheros de configuración.....	123

4.1	AndroidManifest.xml.....	123
4.2	build.gradle (Project).....	124
4.3	build.gradle (App)	125
5	Otros ficheros.....	127
5.1	Ficheros de objetos.....	127
5.1.1	Cuestionario.kt	127
5.1.2	Pregunta.kt	127
5.2	Ficheros de interfaz complementarios	127
5.2.1	Splash.xml.....	127
5.2.2	Bgrn.xml.....	128
5.2.3	Colors.xml.....	128

1 Introducción

En el siguiente documento se mostrará el código realizado para el desarrollo de la aplicación de móvil Quizu. El documento se dividirá en los siguientes apartados:

- **Ficheros de funcionamiento:** En este apartado se mostrará el código de los ficheros Kotlin correspondientes al funcionamiento de la aplicación.
- **Ficheros de interfaz:** Código destinado a la creación de la interfaz y sus elementos. Estos ficheros tienen la extensión .XML y están asociados a un fichero de funcionamiento.
- **Ficheros de configuración:** En este apartado se mostrará el código correspondiente a los ficheros de configuración de la aplicación y los módulos importados.
- **Otros ficheros:** En este apartado se mostrarán los ficheros que no pertenezcan a ninguna de las otras categorías o sean complementarios de ellas.

2 Ficheros de funcionamiento

2.1 AuthActivity.kt

```
package com.example.quizu

import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.view.KeyEvent
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.analytics.FirebaseAnalytics
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_auth.*

class AuthActivity : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        //Guardado de datos para inicio automatico de sesion

        val prefs = getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE)
        val email = prefs.getString("email", null)

        if (email==null)
            setTheme(R.style.Theme_Quizu)
        val analytics:FirebaseAnalytics = FirebaseAnalytics.getInstance(this)
        val bundle = Bundle();
        bundle.putString("mensaje", "Integracion de Firebase Completa")
        analytics.logEvent("InitScreen",bundle)

        session()
        setup()

    }

    private fun setup(){

        setContentView(R.layout.activity_auth)

        registrarButton.setOnClickListener {
```

```

        showSign( "",false)
    }

    accederButton.setOnClickListener {

        if(editTextEmail.text.isNotEmpty() &&
        editTextPassword.text.isNotEmpty()){

            FirebaseAuth.getInstance().signInWithEmailAndPassword(editTextEmail.text.toString(),
            editTextPassword.text.toString()).addOnCompleteListener {

                if (it.isSuccessful){
                    val prefs =
                    getSharedPreferences(getString(R.string.prefs_file),
                    Context.MODE_PRIVATE).edit()

                    prefs.putString("email",editTextEmail.text.toString())
                    prefs.apply()
                    esProfesor()

                }else{
                    showAlert("Email o contraseña incorrecta")
                }
            }

        }

    }

}

fun session(){

    var control=false
    val prefs = getSharedPreferences(getString(R.string.prefs_file),
    Context.MODE_PRIVATE)
    val email = prefs.getString("email", null)

    if (email!=null){

        database.child("Profesores").addValueEventListener(object :
        ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                for(postSnapshot in snapshot.children ){
                    var aux = postSnapshot.child("email").value

                    if (email == aux.toString()) {
                        control=true
                    }
                }
            }
        })
    }
}

```

```

        }
        when(control){
            true->showHome(email)
            false -> showHomeAlumno(email)
        }
    }

    override fun onCancelled(error: DatabaseError) {

    }
})

}

}

private fun esProfesor() {
    database.child("Profesores").addValueEventListener(object :
        ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                for(postSnapshot in snapshot.children){
                    var aux = postSnapshot.child("email").value

                    if (editTextEmail.text.toString() == aux.toString()) {

                        showHome(editTextEmail.text.toString())
                    }
                }
            }

            override fun onCancelled(error: DatabaseError) {

            }
        })
    showHomeAlumno(editTextEmail.text.toString())
}

fun showHome(email: String){
    val homeIntent = Intent(this,HomeActivity::class.java).apply {
        putExtra("email",email)
    }
    startActivity(homeIntent)
}

fun showHomeAlumno(email: String){

```



```

        val intent = Intent(this, HomeAlumno::class.java).apply {
            putExtra("email", email)
        }
        startActivity(intent)
    }

    fun showSign(email: String, checkG: Boolean) {
        val signIntent = Intent(this, Registro::class.java).apply {
            putExtra("email", email)
            putExtra("check", checkG)
        }
        startActivity(signIntent)
    }

    fun showAlert(mensaje: String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Error")
        builder.setMessage(mensaje)
        builder.setPositiveButton("Aceptar", null)
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {
        finishAffinity()
        return super.onKeyDown(keyCode, event)
    }
}

```

2.2 Registro.kt

```

package com.example.quizu

import android.app.VoiceInteractor
import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import com.android.volley.AuthFailureError
import com.android.volley.Request
import com.android.volley.Response
import com.android.volley.VolleyError
import com.android.volley.toolbox.StringRequest
import com.android.volley.toolbox.Volley
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ktx.database
import com.google.firebase.firestore.FirebaseFirestore
import com.google.firebase.ktx.Firebase

```

```

import kotlinx.android.synthetic.main.activity_auth.*
import kotlinx.android.synthetic.main.activity_registro.*
import org.json.JSONArray
import org.json.JSONException
import org.json.JSONObject
import java.util.*
import kotlin.collections.HashMap

class Registro : AppCompatActivity() {

    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_registro)

        val bundle = intent.extras
        val email = bundle?.getString("email")

        setup(email ?: "")
    }

    private fun setup(email:String){

        buttonSignAceptar.setOnClickListener {
            registroNormal()
        }

        buttonSignCancelar.setOnClickListener {

            val prefs = getSharedPreferences(
                getString(R.string.prefs_file),
                Context.MODE_PRIVATE
            ).edit()
            prefs.clear()
            prefs.apply()

            FirebaseAuth.getInstance().signOut()
            onBackPressed()

        }

    }

    fun registroNormal(){

        if (editTextSignEmail.text.isNotEmpty() &&

```

```

editTextSignPassword.text.isNotEmpty() && editTextName.text.isNotEmpty()
    && editTextConfirmSignPassword.text.isNotEmpty() &&
    (radioButtonAlumno.isChecked || radioButtonProfesor.isChecked)
    && editTextDNI.text.isNotEmpty()) {
    if (editTextSignPassword.text.toString() ==
        editTextConfirmSignPassword.text.toString()) {
        FirebaseAuth.getInstance().createUserWithEmailAndPassword(
            editTextSignEmail.text.toString(),
            editTextSignPassword.text.toString()
        ).addOnCompleteListener {
            if (it.isSuccessful) {
                val prefs =
getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE).edit()

prefs.putString("email", editTextSignEmail.text.toString())
prefs.apply()
if (radioButtonAlumno.isChecked) {
    addNewAlumno()

showHomeAlumno(editTextSignEmail.text.toString(), editTextName.text.toString()
)
            }else{
                addNewProfesor()

showHome(editTextSignEmail.text.toString(), editTextName.text.toString())
            }
        }else{
            showAlert("Compruebe que el email introducido es
válido y que la contraseña tiene más de 6 caracteres")
        }
    }
}
}
}
}
}
}
}

private fun addNewAlumno(){
    var name = editTextName.text.toString()
    var email = editTextSignEmail.text.toString()
    var dni = editTextDNI.text.toString()
    var alumno = HashMap<String,String>()

    alumno.put("dni", dni)
    alumno.put("email", email)
    alumno.put("nombre", name)

    database.child("Alumnos").child(dni).setValue(alumno)

```

```

    }

    private fun addNewProfesor(){

        var dni = editTextDNI.text.toString()
        var name = editTextName.text.toString()
        var email = editTextSignEmail.text.toString()
        var profesor = HashMap<String,String>()

        profesor.put("dni", dni)
        profesor.put("email", email)
        profesor.put("nombre", name)

        database.child("Profesores").child(dni).setValue(profesor)

    }

    fun showHome(email:String,nombre:String){
        val homeIntent = Intent(this,HomeActivity::class.java).apply {
            putExtra("email",email)
            putExtra("nombre",nombre)
        }
        startActivity(homeIntent)
    }

    fun showAlert(mensaje:String){
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Error")
        builder.setMessage(mensaje)
        builder.setPositiveButton("Aceptar",null)
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    fun showHomeAlumno(email:String,nombre:String){
        val homeIntent = Intent(this,HomeAlumno::class.java).apply {
            putExtra("email",email)
            putExtra("nombre",nombre)
        }
        startActivity(homeIntent)
    }
}

```

2.3 HombeActivity.kt

```

package com.example.quizu

import android.content.Context
import android.content.DialogInterface
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

```

```

import android.view.KeyEvent
import androidx.appcompat.app.AlertDialog
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_home.*
import kotlinx.android.synthetic.main.activity_home_alumno.*

class HomeActivity : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_home)

        val bundle = intent.extras
        var email = bundle?.getString("email")
        var nombre = bundle?.getString("nombre")

        val prefs = getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE)

        if (email==null){
            email = prefs.getString("email", null)
        }

        setup(email ?: "", nombre ?: "")

    }

    private fun setup(email:String, nombre:String){
        var dni=""

        database.child("Profesores").addValueEventListener(object :
ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    for (postSnapshot in snapshot.children) {
                        if (email ==
postSnapshot.child("email").value.toString()) {
                            dni = postSnapshot.child("dni").value.toString()
                            nametextView.text =
                                postSnapshot.child("nombre").value.toString()
                        }
                    }
                }
            }
        })
    }
}

```

```

        }
    }
}

    override fun onCancelled(error: DatabaseError) {
    }
})

cerrarSesion.setOnClickListener{
    //Borrado de datos cuando cerramos sesion
    val prefs = getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE).edit()
    prefs.clear()
    prefs.apply()

    FirebaseAuth.getInstance().signOut()
    //showLogin()
    finishAffinity()
}

buttonSalirHome.setOnClickListener {
    finishAffinity()
}

buttonEditarPerfil.setOnClickListener {
    showEditarPerfil(dni,"profesor")
}

buttonAsignaturas.setOnClickListener {
    showLista("asignaturasProfesor")
}

buttonCrearAsignatura.setOnClickListener {
    showFormAsignatura()
}

}

private fun showEditarPerfil(dni: String, tipo:String) {
    val intent = Intent(this,editarPerfil::class.java).apply {
        putExtra("dni",dni)
        putExtra("tipo", tipo)
    }
    startActivity(intent)
}

private fun showFormAsignatura() {
    val intent = Intent(this,FormAsignatura::class.java).apply {
    }
    startActivity(intent)
}
}

```

```

private fun showLogin() {
    val intent = Intent(this, AuthActivity::class.java).apply {
        //putExtra("nombre", nombre)
    }
    startActivity(intent)
}

private fun showFormQuiz() {
    val intent = Intent(this, FormQuiz::class.java).apply {
        //putExtra("nombre", nombre)
    }
    startActivity(intent)
}

private fun showLista(listaDe: String) {
    val intent = Intent(this, Lista::class.java).apply {
        putExtra("ListaDe", listaDe)
    }
    startActivity(intent)
}

fun showAlert(mensaje:String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
    builder.setMessage(mensaje)
    builder.setPositiveButton("Aceptar",null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {

    val builder = AlertDialog.Builder(this)
    builder.setTitle("Salir!")
    builder.setMessage("¿Quieres salir de la aplicación")
    builder.setNegativeButton("Volver", DialogInterface.OnClickListener {
dialog, which -> dialog.cancel() })
    builder.setPositiveButton("Salir", DialogInterface.OnClickListener {
dialog, which -> finishAffinity() })
    val dialog: AlertDialog = builder.create()
    dialog.show()
    return false
}

}

```

2.4 HomeAlumno.kt

```

package com.example.quizu

import android.content.Context
import android.content.DialogInterface
import android.content.Intent

```

```

import android.os.Bundle
import android.view.KeyEvent
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_asignatura_alumno.*
import kotlinx.android.synthetic.main.activity_home_alumno.*
import kotlinx.android.synthetic.main.activity_home_alumno.editTextCodigoCuestionarioAlumno

```

```

class HomeAlumno : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rtbd.europe-west1.firebaseio.com/").reference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_home_alumno)

        val bundle = intent.extras
        var email = bundle?.getString("email")
        val nombre = bundle?.getString("nombre")
        val dni = bundle?.getString("dni")

        val prefs = getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE)

        if (email==null){
            email = prefs.getString("email", null)
        }

        setup(email ?: "", nombre ?: "", dni ?: "")
    }

    private fun setup(email: String, nombre: String, dniAlumno: String) {
        var dni=""

        /*if (nombre!="" || nombre!="Nombre") {
            showAlert(nombre)
            nametextViewAlumno.text = nombre
        }*/

        database.child("Alumnos").addValueEventListener(object :
ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    for (postSnapshot in snapshot.children) {
                        if (email ==
postSnapshot.child("email").value.toString()) {

```



```

        dni = postSnapshot.child("dni").value.toString()
        nametextViewAlumno.text =
            postSnapshot.child("nombre").value.toString()
    }
}
}

override fun onCancelled(error: DatabaseError) {
}

})

buttonRealizarCuestionarioHomeAlumno.setOnClickListener {
    var control = false
    database.child("Cuestionarios").addValueEventListener(object :
        ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    for (postSnapshot in snapshot.children) {
                        if (postSnapshot.child("codigo").exists()) {
                            if
(editTextCodigoCuestionarioAlumno.text.toString() == postSnapshot.child(
                                "codigo"
                            ).value.toString()
                        ) {
                            var idQuiz = postSnapshot.key.toString()
                            control = true
                            showSolveQuiz(dni, idQuiz)
                        }
                    }
                }
                if (control == false)
                    showAlert("Clave incorrecta, por favor introduzca
de nuevo")
            }
        })

    override fun onCancelled(error: DatabaseError) {
    }

})

}

buttonAsignaturasAlumno.setOnClickListener{
    showList("asignaturasAlumnos", dni)
}

buttonEditarPerfilAlumno.setOnClickListener {
    showEditarPerfil(dni, "alumno")
}

cerrarSesionHomeAlumno.setOnClickListener {
    val prefs = getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE).edit()

```

```

        prefs.clear()
        prefs.apply()

        FirebaseAuth.getInstance().signOut()
        finishAffinity()
        //showLogin()
    }

    buttonMisAsignaturas.setOnClickListener {
        showList("misAsignaturas", dni)
    }

}

private fun showList(listaDe: String, dni: String) {
    val intent = Intent(this, Lista::class.java).apply {
        putExtra("ListaDe", listaDe)
        putExtra("dni", dni)
    }
    startActivity(intent)
}

private fun showSolveQuiz(dni: String, idQuiz: String) {
    val intent = Intent(this, SolveQuiz::class.java).apply {
        putExtra("dni", dni)
        putExtra("idQuiz", idQuiz)
    }
    startActivity(intent)
}

private fun showLogin() {
    val intent = Intent(this, AuthActivity::class.java).apply {
        //putExtra("nombre", nombre)
    }
    startActivity(intent)
}

private fun showEditarPerfil(dni: String, tipo: String) {
    val intent = Intent(this, editarPerfil::class.java).apply {
        putExtra("dni", dni)
        putExtra("tipo", tipo)
    }
    startActivity(intent)
}

fun showAlert(mensaje: String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
    builder.setMessage(mensaje)
    builder.setPositiveButton("Aceptar", null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

```

```

override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {

    val builder = AlertDialog.Builder(this)
    builder.setTitle("Salir!")
    builder.setMessage("¿Quieres salir de la aplicación")
    builder.setNegativeButton(
        "Volver",
        DialogInterface.OnClickListener { dialog, which ->
            dialog.cancel() })
    builder.setPositiveButton(
        "Salir",
        DialogInterface.OnClickListener { dialog, which ->
            finishAffinity() })
    val dialog: AlertDialog = builder.create()
    dialog.show()
    return false
}
}

```

2.5 AsignaturaAlumno.kt

```

package com.example.quizu

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.KeyEvent
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_asignatura_alumno.*
import kotlinx.android.synthetic.main.activity_asignatura_alumno.editTextCodigoCuestionarioAlumno
import kotlinx.android.synthetic.main.activity_asignatura_profesor.*
import kotlinx.android.synthetic.main.activity_home_alumno.*

class AsignaturaAlumno : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference
    var idQuiz = ""
    var dniA = ""
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_asignatura_alumno)

        val bundle = intent.extras
        val idAsignatura = bundle?.getString("idAsignatura")
        val dni = bundle?.getString("dni")
        dniA = dni.toString()
        setup(idAsignatura ?: "", dni ?: "")
    }
}

```

```

private fun setup(idAsignatura: String, dni: String) {

database.child("Asignaturas").child(idAsignatura).addValueEventListener(object :
t :
    ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            if (snapshot.exists()) {
                textViewNombreAsignaturaAlumno.text=
snapshot.child("nombre").value.toString()

            }
        }
        override fun onCancelled(error: DatabaseError) {

        }
    })

buttonRealizarCuestionarioAlumno.setOnClickListener {
    database.child("Cuestionarios").addValueEventListener(object :
        ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    for (postSnapshot in snapshot.children) {
                        if (postSnapshot.child("codigo").exists()) {
                            if
(editTextCodigoCuestionarioAlumno.text.toString() == postSnapshot.child(
                                "codigo"
                            ).value.toString())
                        ) {
                            var idQuiz = postSnapshot.key.toString()
                            showSolveQuiz(dni, idQuiz)
                        }else{
                            // showAlert("Clave incorrecta, por favor
introduzca de nuevo")
                        }
                    }
                }
            }
            override fun onCancelled(error: DatabaseError) {

            }
        })
}

buttonCuestionariosAlumno.setOnClickListener {
    showList("cuestionariosAlumno",dni)
}

buttonHomeAsignaturaAlumno.setOnClickListener {

```

```

        showHome(dni)
    }

}

private fun showHome(dni: String) {
    val mIntent = Intent(this, HomeAlumno::class.java).apply {
        putExtra("dni", dni)
    }
    startActivity(mIntent)
}

private fun showList(tipo:String,dni: String) {
    val mIntent = Intent(this, Lista::class.java).apply {
        putExtra("ListaDe", tipo)
        putExtra("dni", dni)
    }
    startActivity(mIntent)
}

fun showAlert(mensaje:String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
    builder.setMessage(mensaje)
    builder.setPositiveButton("Aceptar",null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

private fun showSolveQuiz(dni: String, idQuiz: String) {

    val intent = Intent(this, SolveQuiz::class.java).apply {
        putExtra("dni", dni)
        putExtra("idQuiz", idQuiz)
    }
    startActivity(intent)

}

}

```

2.6 AsignaturaProfesor.kt

```

package com.example.quizu

import android.content.DialogInterface
import android.content.Intent
import android.os.Bundle
import android.view.KeyEvent
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError

```

```

import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_asignatura_profesor.*

class AsignaturaProfesor : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference
    var nombre = ""
    var idAsignaturaG=""
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_asignatura_profesor)

        val bundle = intent.extras
        var idAsignatura = bundle?.getString("idAsignatura")
        idAsignaturaG = idAsignatura.toString()
        setup(idAsignatura ?: "")
    }

    private fun setup(idAsignatura: String) {
        textViewIdAsignaturaProfesor.text = idAsignatura

        database.child("Asignaturas").child(idAsignatura).addValueEventListener(object :
            ValueEventListener {
                override fun onDataChange(snapshot: DataSnapshot) {
                    if (snapshot.exists()) {
                        textViewNombreAsignaturaProfesor.text =
                            snapshot.child("nombre").value.toString()
                        nombre = snapshot.child("nombre").value.toString()
                    }
                }

                override fun onCancelled(error: DatabaseError) {
                }
            })

        buttonCrearCuestionarioAsignatura.setOnClickListener {
            showFormQuiz(idAsignatura)
        }

        buttonCuestionariosProfesor.setOnClickListener {
            showListAñadir("cuestionariosProfesor", idAsignatura)
        }

        buttonCuestionariosLanzados.setOnClickListener {
            showList("cuestionariosLanzados", idAsignatura)
        }

        buttonListaAlumnosAsignatura.setOnClickListener {
            showList("alumnos", idAsignatura)
        }
    }
}

```

```
    }

    buttonNotasExportar.setOnClickListener {

        showNotas(idAsignatura,nombre)
    }

    buttonHomeAsignaturaProfesor.setOnClickListener {
        showHome()
    }

    buttonCrearPregutna.setOnClickListener {
        showQuestion(idAsignatura)
    }

    buttonPreguntas.setOnClickListener {
        showListaPreguntas("preguntas", idAsignatura)
    }

}

private fun showListaPreguntas(tipo: String, idAsignatura: String) {
    val mIntent = Intent(this, Lista::class.java).apply {
        putExtra("ListaDe", tipo)
        putExtra("idAsignatura", idAsignatura)
    }
    startActivity(mIntent)
}

private fun showQuestion(idAsignatura: String) {
    val intent = Intent(this, FormQuestion::class.java).apply {
        putExtra("idAsignatura", idAsignatura)
    }
    startActivity(intent)
}

private fun showFormQuiz(idAsignatura: String) {
    val mIntent = Intent(this, FormQuiz::class.java).apply {
        putExtra("idAsignatura", idAsignatura)
    }
    startActivity(mIntent)
}

private fun showHome() {
    val mIntent = Intent(this, HomeActivity::class.java).apply {
        //putExtra("ListaDe",tipo)
    }
    finish()
    startActivity(mIntent)
}
```

```

private fun showNotas(idAsignatura: String, nombre: String) {
    val mIntent = Intent(this, Notas::class.java).apply {
        putExtra("tipo", "asignatura")
        putExtra("idAsignatura", idAsignatura)
        putExtra("nombre", nombre)
    }
    startActivity(mIntent)
}

private fun showList(tipo: String, idAsignatura: String) {
    val mIntent = Intent(this, Lista::class.java).apply {
        putExtra("ListaDe", tipo)
        putExtra("idAsignatura", idAsignatura)
    }
    startActivity(mIntent)
}

private fun showListAñadir(tipo: String, idAsignatura: String) {
    val mIntent = Intent(this, Lista::class.java).apply {
        putExtra("ListaDe", tipo)
        putExtra("idAsignatura", idAsignatura)
    }
    startActivity(mIntent)
}

fun showAlert(mensaje: String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
    builder.setMessage(mensaje)
    builder.setPositiveButton("Aceptar", null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {

    showList("asignaturasProfesor", idAsignaturaG)
    return true
}

}

```

2.7 editarPerfil.kt

```

package com.example.quizu

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_editar_perfil.*
import kotlinx.android.synthetic.main.activity_registro.*

class editarPerfil : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-

```



```

rtdb.europe-west1.firebaseio.com/app/").reference
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_editar_perfil)

        val bundle = intent.extras
        var dni = bundle?.getString("dni")
        var tipo = bundle?.getString("tipo")
        setup(dni?:"", tipo?:"")
    }

    private fun setup(dni: String, tipo:String) {
        buttonCancelarModificarPerfil.setOnClickListener {
            onBackPressed()
        }

        buttonAceptarModificarPerfil.setOnClickListener {
            saveData(dni,tipo)
        }
    }

    private fun saveData(dni: String,tipo:String) {
        var name = editTextName2.text.toString()

        if (tipo == "alumno") {
            database.child("Alumnos").child(dni).child("nombre").setValue(name)
        } else {
            database.child("Profesores").child(dni).child("nombre").setValue(name)
        }
    }

    fun showAlert(mensaje:String){
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Error")
        builder.setMessage(mensaje)
        builder.setPositiveButton("Aceptar",null)
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }
}

```

2.8 FormAsignatura.kt

```

package com.example.quizu

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_form_asignatura.*
import kotlinx.android.synthetic.main.activity_form_quiz.*

```

```

import kotlinx.android.synthetic.main.activity_home.*

class FormAsignatura : AppCompatActivity() {

    var idAsignatura = ""
    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_form_asignatura)

        setup()
    }

    private fun setup() {

        buttonCrearFormAsignatura.setOnClickListener {
            createAsignatura()
            showAsignatura(idAsignatura)
        }

        buttonCancelarFormAsignatura.setOnClickListener {
            onBackPressed()
        }
    }

    private fun showAsignatura(idAsignatura: String) {
        val intent = Intent(this, AsignaturaProfesor::class.java).apply {
            putExtra("idAsignatura", idAsignatura)
        }
        startActivity(intent)
    }

    private fun createAsignatura() {

        var nombre = editTextNombreAsignatura.text.toString()
        var curso = editTextAnoAcademico.text.toString()
        var key = database.push().key
        idAsignatura = key.toString()
        var asignatura = HashMap<String, String>()

        asignatura.put("nombre", nombre)
        asignatura.put("curso", curso)

        if (key != null) {
            database.child("Asignaturas").child(key).setValue(asignatura)
        } else {
            showAlert("Error, se ha generado una clave erronea")
        }
    }

    fun showAlert(mensaje:String){
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Error")
        builder.setMessage(mensaje)
    }
}

```

```

        builder.setPositiveButton("Aceptar",null)
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }
}

```

2.9 FormQuestion.kt

```

package com.example.quizu

import android.content.DialogInterface
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.KeyEvent
import android.view.View
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_form_question.*
import kotlinx.android.synthetic.main.activity_form_quiz.*

class FormQuestion : AppCompatActivity() {

    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference

    private var correctAnswer: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_form_question)

        val bundle = intent.extras
        val numeroPregunta = bundle?.getInt("numeroPreguntas")
        val idQuiz = bundle?.getString("idQuiz")
        val idPregunta = bundle?.getString("idPregunta")
        val idAsignatura = bundle?.getString("idAsignatura")

        setup(numeroPregunta ?: 0, idQuiz ?: "", idPregunta ?: "", idAsignatura
?: "")

    }

    private fun setup(numPregunta: Int, id: String, idPregunta: String,
idAsignatura: String) {

```

```

        controlAnswer()

        if(idPregunta!=""&&idPregunta!=null){
            modificarPregunta(id,idPregunta)
        }

        buttonAnteriorFormQuestion.setOnClickListener {
            controlExit(idAsignatura)
        }

        buttonAceptarFormQuestion.setOnClickListener {
            saveQuestion(id,numPregunta,idPregunta,idAsignatura)
        }
    }
}

private fun saveQuestion(id: String, numPregunta: Int,idPregunta: String,
idAsignatura: String) {
    var question = HashMap<String,String>()
    var numeros = HashMap<String,Long>()
    var control = true
    var numRespuestas:Long = 2
    var correcta = editTextCorrecta.text.toString()
    var key = database.push().key

    if (editTextEnunciadoForm.text.toString()!="") {
        question.put("enunciado", editTextEnunciadoForm.text.toString())
    }else{
        control = false
        showAlert("Introduzca el enunciado de la pregunta")
    }

    if(checkBoxOpcion1.isChecked&& editTextFromOpcion1.text.toString() !=
    "") {
        question.put("respuesta1", editTextFromOpcion1.text.toString())
    }else{
        control = false
        showAlert("La pregunta debe contener al menos dos posibles
respuestas")
    }

    if(checkBoxOpcion2.isChecked&& editTextFromOpcion2.text.toString() !=
    "") {
        question.put("respuesta2", editTextFromOpcion2.text.toString())
        if (correcta=="3" ||correcta=="4"){
            showAlert("La respuesta correcta no debe ser mayor que 2")
            control = false
        }
    }else{
        control = false
        showAlert("La pregunta debe contener al menos dos posibles
respuestas")
    }
}

```

```

    }

    if (checkBoxOpcion3.isChecked && editTextFromOpcion3.text.toString()
!= "") {
        question.put("respuesta3", editTextFromOpcion3.text.toString())

        numRespuestas=3

        if (correcta=="4"){
            showAlert("La respuesta correcta no debe ser mayor que 3")
            control = false
        }
    }

    if (checkBoxOpcion4.isChecked && editTextFromOpcion4.text.toString()
!= "") {
        question.put("respuesta4", editTextFromOpcion4.text.toString())
        numRespuestas=4
    }
    if ((correcta=="1" || correcta=="2" || correcta=="3" ||
correcta=="4")){
        // if (correcta.toLong() <= numRespuestas && correcta.toLong()>0)
    {
        question.put("correcta", editTextCorrecta.text.toString())
        //numeros.put("correcta", correcta.toLong())

    }else{
        showAlert("Introduzca un número entre 1 y 4")
        control = false
    }
    numeros.put("numeroRespuestas", numRespuestas)

    if (control) {
        //Si se le pasa id de cuestionario
        if (id != "") {
            if (key != null) {
                //Si se le pasa el id de pregunta
                if (idPregunta!=""&&idPregunta!=null) {

database.child("Cuestionarios").child(id).child("Preguntas").child(idPregunta
)

                    .setValue(question+numeros)

database.child("Preguntas").child(idPregunta).setValue(question+numeros)
                    database.child("Asignaturas-
Preguntas").child(idAsignatura).child(idPregunta).setValue(idPregunta)

                }else{

database.child("Cuestionarios").child(id).child("Preguntas").child(key)
                    .setValue(question+numeros)

database.child("Preguntas").child(key).setValue(question+numeros)
                    database.child("Asignaturas-

```

```

Preguntas").child(idAsignatura).child(key).setValue(key)
        }
    }
    } //Si no se Le pasa id de cuestionario
    else if (key != null) {

database.child("Preguntas").child(key).setValue(question+numeros)
        database.child("Asignaturas-
Preguntas").child(idAsignatura).child(key).setValue(key)
    }
    if (id != "" || id == null) {
        showFormQuiz(id,idAsignatura)
    } else {
        showAsignatura(idAsignatura)
    }
}
}

private fun modificarPregunta(idQuiz: String,idPregunta: String) {

database.child("Cuestionarios").child(idQuiz).child("Preguntas").child(idPreg
unta).addListenerForSingleValueEvent(object :
    ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            if (snapshot.exists()) {
                var numeroRespuestas =
snapshot.child("numeroRespuestas").value.toString()

                when {
                    numeroRespuestas == "4" -> {

editTextFromOpcion1.setText(snapshot.child("Respuesta1").value.toString())
editTextFromOpcion2.setText(snapshot.child("Respuesta2").value.toString())
editTextFromOpcion3.setText(snapshot.child("Respuesta3").value.toString())
editTextFromOpcion4.setText(snapshot.child("Respuesta4").value.toString())
editTextEnunciadoForm.setText(snapshot.child("enunciado").value.toString())

editTextPenalty.setText(snapshot.child("correcta").value.toString())
                    checkBoxOpcion3.isChecked = true
                    checkBoxOpcion4.isChecked = true
                    checkBoxOpcion3.visibility = View.VISIBLE
                    checkBoxOpcion4.visibility = View.VISIBLE
                    editTextFromOpcion3.visibility = View.VISIBLE
                    editTextFromOpcion4.visibility = View.VISIBLE

                }
                numeroRespuestas == "3" -> {

editTextFromOpcion1.setText(snapshot.child("Respuesta1").value.toString())
editTextFromOpcion2.setText(snapshot.child("Respuesta2").value.toString())

```

```

editTextFromOpcion3.setText(snapshot.child("Respuesta3").value.toString())

editTextEnunciadoForm.setText(snapshot.child("enunciado").value.toString())

editTextPenalty.setText(snapshot.child("correcta").value.toString())
    checkBoxOpcion3.isChecked = true
    checkBoxOpcion3.visibility = View.VISIBLE
    editTextFromOpcion3.visibility = View.VISIBLE
    checkBoxOpcion4.visibility = View.VISIBLE
}

else -> {

editTextFromOpcion1.setText(snapshot.child("Respuesta1").value.toString())

editTextFromOpcion2.setText(snapshot.child("Respuesta2").value.toString())

editTextEnunciadoForm.setText(snapshot.child("enunciado").value.toString())

editTextPenalty.setText(snapshot.child("correcta").value.toString())
    checkBoxOpcion3.visibility = View.VISIBLE

}
}

```

```

    }
}

override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ")
}

})
}

```

```

private fun showAsignatura(idAsignatura: String) {
    val homeIntent = Intent(this,AsignaturaProfesor::class.java).apply {
        putExtra("idAsignatura",idAsignatura)
    }
    startActivity(homeIntent)
}

private fun showFormQuiz(idQuiz:String,idAsignatura: String) {
    val homeIntent = Intent(this,FormQuiz::class.java).apply {
        putExtra("idQuiz",idQuiz)
        putExtra("idAsignatura",idAsignatura)
    }
    startActivity(homeIntent)
}

```

```

fun showAlert(mensaje:String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
    builder.setMessage(mensaje)
    builder.setNegativeButton("Aceptar" , DialogInterface.OnClickListener
{ dialog, which -> dialog.cancel() })
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

```

```

private fun controlAnswer() {
    var control1 = 1.0
    var control2 = 1.0
    var control3 = 0.0
    var control4 = 0.0

    // editTextFromOpcion1.visibility = View.INVISIBLE
    //editTextFromOpcion2.visibility = View.INVISIBLE
    editTextFromOpcion3.visibility = View.INVISIBLE
    editTextFromOpcion4.visibility = View.INVISIBLE

    checkBoxOpcion1.isChecked = true
    checkBoxOpcion2.isChecked = true
    // checkBoxOpcion3.visibility = View.INVISIBLE
    checkBoxOpcion4.visibility = View.INVISIBLE

    checkBoxOpcion1.setOnClickListener {
        if(control1%2==0.0) {
            editTextFromOpcion1.visibility = View.VISIBLE
            checkBoxOpcion2.visibility = View.VISIBLE
            control1++
        }else{
            editTextFromOpcion1.visibility = View.INVISIBLE
            editTextFromOpcion2.visibility = View.INVISIBLE
            editTextFromOpcion3.visibility = View.INVISIBLE
            editTextFromOpcion4.visibility = View.INVISIBLE

            checkBoxOpcion2.visibility = View.INVISIBLE
            checkBoxOpcion3.visibility = View.INVISIBLE
            checkBoxOpcion4.visibility = View.INVISIBLE

            checkBoxOpcion2.isChecked = false
            checkBoxOpcion3.isChecked = false
            checkBoxOpcion4.isChecked = false
            control1 =0.0
            control2 = 0.0
            control3 = 0.0
            control4 = 0.0
        }
    }
}

```

```

checkBoxOpcion2.setOnClickListener {
    if(control2%2==0.0){
        editTextFromOpcion2.visibility = View.VISIBLE
        checkBoxOpcion3.visibility = View.VISIBLE
    }
}

```



```

        control2++
    }else{
        editTextFromOpcion2.visibility = View.INVISIBLE
        editTextFromOpcion3.visibility = View.INVISIBLE
        editTextFromOpcion4.visibility = View.INVISIBLE

        checkBoxOpcion3.visibility = View.INVISIBLE
        checkBoxOpcion4.visibility = View.INVISIBLE

        checkBoxOpcion3.isChecked = false
        checkBoxOpcion4.isChecked = false
        control2=0.0
        control3 = 0.0
        control4 = 0.0
    }
}

checkBoxOpcion3.setOnClickListener {
    if (control3%2==0.0){
        editTextFromOpcion3.visibility = View.VISIBLE
        checkBoxOpcion4.visibility = View.VISIBLE
        control3++
    }else{
        editTextFromOpcion3.visibility = View.INVISIBLE
        editTextFromOpcion4.visibility = View.INVISIBLE

        checkBoxOpcion4.visibility = View.INVISIBLE

        checkBoxOpcion4.isChecked = false
        control3=0.0
        control4 = 0.0
    }
}

checkBoxOpcion4.setOnClickListener {
    if (control4%2==0.0){
        editTextFromOpcion4.visibility = View.VISIBLE
        control4++
    }else{
        editTextFromOpcion4.visibility = View.INVISIBLE
        control4=0.0
    }
}

}
private fun controlExit(idAsignatura: String) {
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Cuidado!")
    builder.setMessage("No has completado la pregunta, ¿seguro que
quieres cancelar?")
    builder.setNegativeButton("Volver a la pregunta",
DialogInterface.OnClickListener { dialog, which -> dialog.cancel() })
    builder.setPositiveButton("Salir", DialogInterface.OnClickListener {

```

```

dialog, which ->
    showAsignatura(idAsignatura)
})
val dialog: AlertDialog = builder.create()
dialog.show()
}
}

```

2.10 FormQuiz.kt

```

package com.example.quizu

import android.content.DialogInterface
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.KeyEvent
import android.widget.ArrayAdapter
import androidx.appcompat.app.AlertDialog
import androidx.core.view.size
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_form_quiz.*

class FormQuiz : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rttdb.europe-west1.firebaseio.com/").reference
    var quiz = ""
    var asig = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_form_quiz)

        val bundle = intent.extras
        val id = bundle?.getString("idQuiz")
        val idAsignatura = bundle?.getString("idAsignatura")

        if (id != null) {
            quiz = id
        }
        if (idAsignatura != null) {
            asig = idAsignatura
        }

        setup(id ?: "", idAsignatura ?: "")
    }

    private fun setup(id: String, idAsignatura: String) {

```

```

var idQuiz= ""

if (id!=""){
    idQuiz = id

database.child("Cuestionarios").child(idQuiz).addValueEventListener(object :
 ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()) {
            var titulo = snapshot.child("Titulo").value
            var pen = snapshot.child("Penalizacion").value

            editTextPenalty.setText(pen.toString())
            editTextTituloQuiz.setText(titulo.toString())
        }
    }

    override fun onCancelled(error: DatabaseError) {
    }
})

}else if(id==""){

    idQuiz = QuizcreateIdQuiz()

}

var aununciado = arrayListOf<String>()
var numRes = arrayListOf<String>()
var aopcion1 = arrayListOf<String>()
var aopcion2 = arrayListOf<String>()
var aopcion3 = arrayListOf<String>()
var aopcion4 = arrayListOf<String>()
var acorrecta = arrayListOf<String>()
var aid = arrayListOf<String>()
var adapter =
ArrayAdapter<this, android.R.layout.simple_list_item_1, aununciado>

database.child("Cuestionarios").child(idQuiz).child("Preguntas").addListenerF
orSingleValueEvent(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()) {
            for (postSnapshot in snapshot.children) {
                var enunciado =
postSnapshot.child("enunciado").value
                var numeroRespuestas =
postSnapshot.child("numeroRespuestas").value
                var opcion1 =
postSnapshot.child("respuesta1").value
                var opcion2 =
postSnapshot.child("respuesta2").value
                var opcion3 =

```

```

postSnapshot.child("respuesta3").value
    var opcion4 =
postSnapshot.child("respuesta4").value
    var correcta =
postSnapshot.child("correcta").value
    var idAsignatura = postSnapshot.key

    numRes.add(numeroRespuestas.toString())
    aopcion1.add(opcion1.toString())
    aopcion2.add(opcion2.toString())
    aopcion3.add(opcion3.toString())
    aopcion4.add(opcion4.toString())
    acorrecta.add(correcta.toString())
    aenunciado.add(enunciado.toString())
    aid.add(idAsignatura.toString())

    listPreguntasQuiz.adapter = adapter

        }
    }
}

override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ")
}

})

    listPreguntasQuiz.setOnItemClickListener { parent, view, position,
id ->
        showSeleccionado(aid[position],idQuiz,idAsignatura)
    }

    buttonCancelarFormQuiz.setOnClickListener {
        controlExit(idQuiz,idAsignatura)
    }

    buttonAñadirPreguntaFormQuiz.setOnClickListener {

database.child("Cuestionarios").child(idQuiz).child("Penalizacion").setValue(
editTextPenalty.text.toString())

database.child("Cuestionarios").child(idQuiz).child("Titulo").setValue(editTe
xtTituloQuiz.text.toString())
        showList(idQuiz,"preguntas",idAsignatura)
    }

    buttonCrearPreguntaFormQuiz.setOnClickListener{

database.child("Cuestionarios").child(idQuiz).child("Penalizacion").setValue(
editTextPenalty.text.toString())

```

```

database.child("Cuestionarios").child(idQuiz).child("Titulo").setValue(editTextTituloQuiz.text.toString())
        showFormQuestion(idQuiz,idAsignatura)
    }

    buttonCrearFormQuiz.setOnClickListener {

        if(editTextTituloQuiz.text.isEmpty()||editTextPenalty.toString().isEmpty()) {
            showAlert("Introduzca valores correctos a todos los campos")
        }else if(listPreguntasQuiz.size==0){
            showAlert("No se ha introducido ninguna pregunta")
        }else{
            crearCuestionario(idQuiz,idAsignatura)
            showAsignatura(idAsignatura)
        }

    }

}

override fun onBackPressed() {
    controlExit(quiz,asig)
}

private fun crearCuestionario(idQuiz: String, idAsignatura: String) {
    var titulo = editTextTituloQuiz.text.toString()
    database.child("Asignaturas-Cuestionarios").child(idAsignatura).child(idQuiz).child("Titulo").setValue(titulo)
    database.child("Asignaturas-Cuestionarios").child(idAsignatura).child(idQuiz).setValue(idQuiz)
}

private fun QuizcreateIdQuiz(): String {
    var key = database.push().key
    return key.toString()
}

private fun borrarQuiz(idQuiz: String) {
    var key = idQuiz
    database.child("Cuestionarios").child(key).removeValue()
}

private fun showFormQuestion(id:String,idAsignatura: String) {
    val homeIntent = Intent(this,FormQuestion::class.java).apply {
        putExtra("idQuiz",id)
        putExtra("idAsignatura",idAsignatura)
    }
    startActivity(homeIntent)
}

fun showAlert(mensaje:String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
}

```

```

        builder.setMessage(mensaje)
        builder.setPositiveButton("Aceptar",null)
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }
    private fun showSeleccionado(seleccionado: String, idQuiz:
String,idAsignatura: String) {
        val mintent = Intent(this,Seleccionado::class.java).apply {
            putExtra("idQuiz",idQuiz)
            putExtra("idPregunta",seleccionado)
            putExtra("tipoSelect","preguntas")
            putExtra("idAsignatura",idAsignatura)

        }
        startActivity(mintent)
    }

    private fun showList(idQuiz: String, tipo: String, idAsignatura: String)
{
        val mIntent = Intent(this,Lista::class.java).apply {
            putExtra("idQuiz",idQuiz)
            putExtra("ListaDe",tipo)
            putExtra("idAsignatura",idAsignatura)
        }
        startActivity(mIntent)
    }

    private fun controlExit(idQuiz: String,idAsignatura: String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
        builder.setMessage("No has completado el cuestionario, ¿seguro que
quieres cancelar?")
        builder.setNegativeButton("Volver al cuestionario",
DialogInterface.OnClickListener { dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Salir", DialogInterface.OnClickListener {
dialog, which ->
            borrarQuiz(idQuiz)
            showAsignatura(idAsignatura)
        })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun showAsignatura(idAsignatura: String) {
        val mIntent = Intent(this,AsignaturaProfesor::class.java).apply {
            putExtra("idAsignatura",idAsignatura)
        }
        startActivity(mIntent)
    }

    private fun showHome() {
        val mIntent = Intent(this,HomeActivity::class.java).apply {

        }
        startActivity(mIntent)
    }
}

```

2.11 Lista.kt

```
package com.example.quizu

import android.content.DialogInterface
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.KeyEvent
import android.view.View
import android.widget.AdapterView
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_lista.*

class Lista : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rttdb.europe-west1.firebaseio.com/").reference
    var pos = -1
    var control = false
    var tipoLista = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_lista)

        val bundle = intent.extras
        val tipo = bundle?.getString("ListaDe")
        val idQuiz = bundle?.getString("idQuiz")
        val idAsignatura = bundle?.getString("idAsignatura")
        val idAlumno = bundle?.getString("idAlumno")
        val dni = bundle?.getString("dni")

        tipoLista = tipo.toString()

        setup(tipo ?: "", idQuiz ?: "", idAsignatura ?: "", idAlumno ?: "", dni
?: "")

    }

    private fun setup(tipo: String, idQuiz: String, idAsignatura: String,
idAlumno:String,dni: String) {

        buttonNotas.visibility = View.INVISIBLE

        var adapter :ArrayAdapter<String>

        if (tipo == "preguntas") {
            buttonSeleccionar.visibility = View.INVISIBLE
        }
    }
}
```

```

        textViewLista.text = "Lista de preguntas: "
        var aenunciado = arrayListOf<String>()
        var numRes = arrayListOf<String>()
        var aopcion1 = arrayListOf<String>()
        var aopcion2 = arrayListOf<String>()
        var aopcion3 = arrayListOf<String>()
        var aopcion4 = arrayListOf<String>()
        var acorrecta = arrayListOf<String>()
        var aIdpregunta = arrayListOf<String>()

        adapter =
            ArrayAdapter<this, android.R.layout.simple_list_item_1, aenunciado>

            database.child("Asignaturas-
Preguntas").child(idAsignatura).addValueEventListener(object
: ValueEventListener {
                override fun onDataChange(psnapshot: DataSnapshot) {
                    if (psnapshot.exists()) {
                        for (postSnapshot in psnapshot.children) {
                            var id = postSnapshot.key

database.child("Preguntas").child(id.toString()).addValueEventListener(object
: ValueEventListener {
                            override fun onDataChange(snapshot:
DataSnapshot) {

                                    if (snapshot.exists()) {

                                            var enunciado =
snapshot.child("enunciado").value
                                            var numeroRespuestas =
snapshot.child("numeroRespuestas").value
                                            var opcion1 =
snapshot.child("respuesta1").value
                                            var opcion2 =
snapshot.child("respuesta2").value
                                            var opcion3 =
snapshot.child("respuesta3").value
                                            var opcion4 =
snapshot.child("respuesta4").value
                                            var correcta =
snapshot.child("correcta").value
                                            var idPregunta = snapshot.key

numRes.add(numeroRespuestas.toString())
aopcion1.add(opcion1.toString())
aopcion2.add(opcion2.toString())
aopcion3.add(opcion3.toString())
aopcion4.add(opcion4.toString())

acorrecta.add(correcta.toString())

aenunciado.add(enunciado.toString())

aIdpregunta.add(idPregunta.toString())

```



```

        listView.adapter = adapter
    }
}
override fun onCancelled(error:
DatabaseError) {
    showAlert("Error al acceder a la base de
datos ", ":")
}
})
}
}
}
override fun onCancelled(error: DatabaseError) {
}
})

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position
    if(idQuiz==""){
showSeleccionarPregunta("preguntasBanco",aIdpregunta[pos],idAsignatura)
    }else {
        if (pos >= 0) {
            when {
                aopcion4[pos] != "null" -> {

database.child("Cuestionarios").child(idQuiz).child("Preguntas")
                .child(aIdpregunta[pos]).child("enunciado")
                    .setValue(aenunciado[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")
                .child(aIdpregunta[pos]).child("respuesta1")
                    .setValue(aopcion1[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")
                .child(aIdpregunta[pos]).child("respuesta2")
                    .setValue(aopcion2[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")
                .child(aIdpregunta[pos]).child("respuesta3")
                    .setValue(aopcion3[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")
                .child(aIdpregunta[pos]).child("respuesta4")
                    .setValue(aopcion4[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

```

```

        .child(aIdpregunta[pos]).child("correcta")
            .setValue(acorrecta[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("numeroRespuestas")
            .setValue(numRes[pos])

    }
    aopcion3[pos] != "null" -> {

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("enunciado")
            .setValue(aenunciado[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("respuesta1")
            .setValue(aopcion1[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("respuesta2")
            .setValue(aopcion2[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("respuesta3")
            .setValue(aopcion3[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("correcta")
            .setValue(acorrecta[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("numeroRespuestas")
            .setValue(numRes[pos])

    }
    else -> {

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("enunciado")
            .setValue(aenunciado[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("respuesta1")
            .setValue(aopcion1[pos])

database.child("Cuestionarios").child(idQuiz).child("Preguntas")

```

```

        .child(aIdpregunta[pos]).child("respuesta2")
            .setValue(aopcion2[pos])

        database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("correcta")
            .setValue(acorrecta[pos])

        database.child("Cuestionarios").child(idQuiz).child("Preguntas")

        .child(aIdpregunta[pos]).child("numeroRespuestas")
            .setValue(numRes[pos])

    }

    }

    showFormQuiz(idQuiz, idAsignatura)
}

}

}

if (tipo == "asignaturasProfesor") {
    buttonSeleccionar.visibility = View.INVISIBLE
    textViewLista.text = "Lista de asignaturas: "
    var anombre = arrayListOf<String>()
    var acurso = arrayListOf<String>()
    var aid = arrayListOf<String>()

    adapter = ArrayAdapter<this, android.R.layout.simple_list_item_1,
anombre>

    database.child("Asignaturas").addValueEventListener(object :
ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {

            if (snapshot.exists()) {
                for (postSnapshot in snapshot.children) {
                    var nombre = postSnapshot.child("nombre").value
                    var curso = postSnapshot.child("curso").value

                    var idAsignatura = postSnapshot.key

                    anombre.add(nombre.toString())
                    acurso.add(curso.toString())
                    aid.add(idAsignatura.toString())

                    listView.adapter = adapter
                }
            }
        }
    })
}

```

```

    }

    override fun onCancelled(error: DatabaseError) {
        showAlert("Error al acceder a la base de datos ", ":")
    }
})

listView.adapter = adapter

listView.setOnItemClickListener { parent, view, position, id ->
    showAsignatura(aid[position])
}

}

if(tipo == "alumnos") {
    var idA = idAsignatura
    buttonSeleccionar.text = "Notas"
    buttonSeleccionar.visibility = View.INVISIBLE
    textViewLista.text = "Lista de alumnos: "
    var anombre = arrayListOf<String>()
    var aemail = arrayListOf<String>()
    var adni = arrayListOf<String>()

    var aidAlumnos = arrayListOf<String>()
    adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
anombre)
    database.child("Asignaturas-
Alumnos").child(idA).addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {

            if (snapshot.exists()) {
                for (postSnapshot in snapshot.children) {

                    var dni = postSnapshot.child("dni").value

                    database.child("Alumnos").child(dni.toString()).addValueEventListener(object
: ValueEventListener {
                        override fun onDataChange(snapshot:
DataSnapshot) {

                            if (snapshot.exists()) {

                                var dnid =
snapshot.child("dni").value
                                var nombre =
snapshot.child("nombre").value
                                var email =
snapshot.child("email").value

                                var idAlumno = snapshot.key

                                anombre.add(nombre.toString())
                                aemail.add(email.toString())
                                adni.add(dnid.toString())

                                aidAlumnos.add(idAlumno.toString())

                                listView.adapter = adapter

```

```

        }
    }

    override fun onCancelled(error:
DatabaseError) {
        showAlert("Error al acceder a la base de
datos ", ":")
    }
})

}

}

}

override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ", ":")
}

})

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position
    textViewDesplegadoLista.text = "DNI: " + adni[position] +
        System.getProperty("line.separator") +
        "Nombre: " + anombre[position] +
        System.getProperty("line.separator") +
        "Email: " + aemail[position]
    buttonSeleccionar.visibility = View.VISIBLE
}

buttonSeleccionar.setOnClickListener {
    showListaNotas("notasProfesor",idAsignatura,adni[pos])
}

}

if(tipo == "cuestionariosProfesor") {
    buttonNotas.visibility = View.INVISIBLE
    buttonSeleccionar.visibility = View.INVISIBLE
    var idA= idAsignatura
    textViewLista.text = "Lista de cuestionarios: "
    var atitulo = arrayListOf<String>()
    var apenalty = arrayListOf<String>()
    var aidCuestionarios = arrayListOf<String>()
    var aPreguntas = arrayListOf<String>()

    adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
atitulo)

    database.child("Asignaturas-
Cuestionarios").child(idA).addValueEventListener(object : ValueEventListener
{
        override fun onDataChange(snapshot: DataSnapshot) {
            if (snapshot.exists()) {
                for (postSnapshot in snapshot.children) {

```

```

        var idCuestionario = postSnapshot.key

database.child("Cuestionarios").child(idCuestionario.toString()).addValueEventListener(object : ValueEventListener {
    override fun onDataChange(snapshot:
DataSnapshot) {
        if (snapshot.exists()) {
            var titulo =
            snapshot.child("Titulo").value
            var penalty =
            snapshot.child("Penalizacion").value
            var idCuestionario = snapshot.key
            var pregunta =
            snapshot.child("Preguntas")

            atitulo.add(titulo.toString())
            apenalty.add(penalty.toString())

            aidCuestionarios.add(idCuestionario.toString())
            aPreguntas.add(pregunta.toString())

            listView.adapter = adapter
        }
    }
    override fun onCancelled(error:
DatabaseError) {
        showAlert("Error al acceder a la base de
datos ", ":")
    }
})
})

override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ", ":")
}
})

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position

showSeleccionar("cuestionarios",aidCuestionarios[pos],idAsignatura)

    }

    buttonNotas.setOnClickListener {

showListaNotasAsignatura("notasCuestionarios",idAsignatura,aidCuestionarios[pos])

    }

    buttonSeleccionar.setOnClickListener {

```

```

showSeleccionar("cuestionarios",aidCuestionarios[pos],idAsignatura)
    }
}

if(tipo == "añadirCuestionarios") {
    buttonSeleccionar.text="Añadir"
    textViewLista.text = "Lista de cuestionarios: "
    var atitulo = arrayListOf<String>()
    var apenalty = arrayListOf<String>()
    var aidCuestionarios = arrayListOf<String>()

    adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
atitulo)

        database.child("Cuestionarios").addValueEventListener(object
: ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    for (postSnapshot in snapshot.children) {
                        var titulo =
postSnapshot.child("Titulo").value
                        var penalty =
postSnapshot.child("Penalizacion").value
                        var idCuestionario = postSnapshot.key

                        atitulo.add(titulo.toString())
                        apenalty.add(penalty.toString())

                        aidCuestionarios.add(idCuestionario.toString())

                    }
                }
            }

            override fun onCancelled(error: DatabaseError) {
                showAlert("Error al acceder a la base de datos ",
":")
            }
        })

    listView.adapter = adapter

    listView.setOnItemClickListener { parent, view, position, id ->
        pos = position
        textViewDesplegadoLista.text = "Titulo: " + atitulo[position]
+
        System.getProperty("line.separator") +
        "Penalizacion al fallo: " + apenalty[position]

    }

    buttonSeleccionar.setOnClickListener {
        añadirCuestionario(idAsignatura,aidCuestionarios[pos])
    }
}

```

```

        if(tipo == "notasProfesor") {
            buttonSeleccionar.text="Nota Media"
            var idA= idAsignatura
            var dni = idAlumno
            textViewLista.text = "Lista de cuestionarios: "

            var anotas = arrayListOf<String>()
            var atitulo = arrayListOf<String>()
            adapter = ArrayAdapter<this, android.R.layout.simple_list_item_1,
atitulo>

                database.child("Asignaturas-
Cuestionarios").child(idA).addValueEventListener(object : ValueEventListener
{
                    override fun onDataChange(snapshot: DataSnapshot) {
                        if (snapshot.exists()) {
                            for (postSnapshot in snapshot.children) {
                                var idCuestionario = postSnapshot.key
                                database.child("Alumnos-
Cuestionarios").child(dni).addValueEventListener(object : ValueEventListener
{
                                    override fun onDataChange(snapshot:
DataSnapshot) {
                                        if (snapshot.exists()) {

                                            for (xpostSnapshot in
snapshot.children) {
                                                if (xpostSnapshot.exists()) {

                                                    if
(idCuestionario.toString() == xpostSnapshot.key.toString()) {

                                                        var nota =
xpostSnapshot.child("nota").value
                                                        var titulo =
xpostSnapshot.child("titulo").value

                                                        anotas.add(nota.toString())
                                                        atitulo.add(titulo.toString())

                                                    }

                                                }

                                            }

                                        }

                                    }

                                }

                            }

                        }

                    }

                }

            listView.adapter =
adapter
        }
    }
}

```



```

                                override fun onCancelled(error:
DatabaseError) {
                                showAlert("Error al acceder a la base de
datos ", ":")
                                }
                                })
                                }
                                }
                                }

                                override fun onCancelled(error: DatabaseError) {
                                showAlert("Error al acceder a la base de datos ", ":")
                                }
                                })

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position
    textViewDesplegadoLista.text = "Titulo: " + atitulo[position]
+
    System.getProperty("line.separator") +
    "Nota: " + anotas[position]

}

buttonSeleccionar.setOnClickListener {
    var notaMedia = 0.0
    for(i in anotas)
        notaMedia += i.toDouble()

    notaMedia /= anotas.size
    textViewNotaMedia.text = "Nota media en los cuestionarios: "+
notaMedia.toString()
}

}

if(tipo == "notasCuestionarios") {
    buttonNotas.visibility=View.VISIBLE
    buttonNotas.text = "Nota media"
    var idA= idAsignatura
    var idCuestionario = idQuiz
    textViewLista.text = "Seleccione un cuestionario: "

    var anombre = arrayListOf<String>()
    var anotas = arrayListOf<String>()
    var adni = arrayListOf<String>()

    adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
adni)
    database.child("Alumnos-
Cuestionarios").addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            if (snapshot.exists()) {
                for (postSnapshot in snapshot.children) {

```

```

        for (ppostSnapshot in postSnapshot.children) {
            if(ppostSnapshot.key.toString() ==
idCuestionario) {
                var id = postSnapshot.key
                var nota =
ppostSnapshot.child("nota").value
                var nombre =
postSnapshot.child("nombre").value

                anotas.add(nota.toString())
                adni.add(id.toString())
                anombre.add(nombre.toString())

                listView.adapter = adapter
            }
        }
    }
}

override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ", ":")
}

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position
    textViewDesplegadoLista.text = "DNI: " + adni[position] +
        System.getProperty("line.separator") +
        "Nombre: " + anombre[position] +
        System.getProperty("line.separator") +
        "Nota: " + anotas[position]
}

buttonNotas.setOnClickListener {
    var notaMedia = 0.0
    for(i in anotas)
        notaMedia += i.toDouble()

    notaMedia /= anotas.size
    textViewDesplegadoLista.text = "Nota media de los alumnos: " +
notaMedia.toString()
}

}

if(tipo == "cuestionariosLanzados") {
    var idA= idAsignatura

    textViewLista.text = "Lista de cuestionarios: "
    buttonSeleccionar.text = "Cerrar cuestionario"
    buttonSeleccionar.visibility = View.INVISIBLE

    var atitulo = arrayListOf<String>()
    var acodigo = arrayListOf<String>()
    var aid = arrayListOf<String>()

```

```
adapter = ArrayAdapter<this>, android.R.layout.simple_list_item_1,
atitulo)
        database.child("Asignaturas-
Cuestionarios").child(idA).addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    for (postSnapshot in snapshot.children) {
                        var idCuestionario = postSnapshot.key

database.child("Cuestionarios").child(idCuestionario.toString())
                            .addValueEventListener(object :
ValueEventListener {
                                override fun onDataChange(snapshot:
DataSnapshot) {
                                    if (snapshot.exists()) {
                                        if (snapshot.child("codigo")
                                            .exists() //&&
snapshot.child("codigo").value.toString() != ""
                                                ) {
                                                    var id = snapshot.key
                                                    var codigo =
snapshot.child("codigo").value
                                                    var titulo =
snapshot.child("Titulo").value
atitulo.add(titulo.toString())
aid.add(id.toString())
acodigo.add(codigo.toString())

listView.adapter =
adapter
}
    }
})
override fun onCancelled(error:
DatabaseError) {
    showAlert("Error al acceder a la base
de datos ", ":")
}
}))
}
}
override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ", ":")
}
}))
```

```

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position
    buttonSeleccionar.visibility = View.VISIBLE
    textViewDesplegadoLista.text = "ID: " + aid[position] +
        System.getProperty("line.separator") +
        "Titulo: " + atitulo[position] +
        System.getProperty("line.separator") +
        "Codigo: " + acodigo[position]
}

buttonSeleccionar.setOnClickListener {
    cerrarCuestionario(aid[pos], idAsignatura)
}

}

if(tipo == "asignaturasAlumnos") {
    var idni = dni
    buttonSeleccionar.visibility = View.INVISIBLE
    buttonSeleccionar.text = "Matricularse"
    textViewLista.text = "Lista de asignaturas: "
    var anombre = arrayListOf<String>()
    var acurso = arrayListOf<String>()
    var aid = arrayListOf<String>()

    adapter = ArrayAdapter<this, android.R.layout.simple_list_item_1,
anombre>

    database.child("Asignaturas").addValueEventListener(object :
ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {

            if (snapshot.exists()) {
                for (postSnapshot in snapshot.children) {
                    var nombre = postSnapshot.child("nombre").value
                    var curso = postSnapshot.child("curso").value

                    var idAsignatura = postSnapshot.key

                    anombre.add(nombre.toString())
                    acurso.add(curso.toString())
                    aid.add(idAsignatura.toString())

                    listView.adapter = adapter
                }
            }

            override fun onCancelled(error: DatabaseError) {
                showAlert("Error al acceder a la base de datos ", ":")
            }
        })

listView.setOnItemClickListener { parent, view, position, id ->

```

```

        pos = position

        database.child("Asignaturas-
Alumnos").child(aid[pos]).child(dni).addValueEventListener(object :
ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {
                    showAsignaturaAlumno(aid[pos], dni)

                } else {

                    buttonSeleccionar.visibility = View.VISIBLE
                    textViewDesplegadoLista.text = "Nombre: " +
anombre[pos] +
                        System.getProperty("line.separator") +
                        "Curso: " + acurso[pos]

                }
            }

            override fun onCancelled(error: DatabaseError) {
                showAlert("Error al acceder a la base de datos ",
":")
            }
        })
    }

    buttonSeleccionar.setOnClickListener {
        matricularAlumno(aid[pos],dni)
        showAsignaturaAlumno(aid[pos],dni)
    }

}

if(tipo == "misAsignaturas") {

    buttonSeleccionar.visibility = View.INVISIBLE

    textViewLista.text = "Lista de asignaturas: "
    var anombre = arrayListOf<String>()
    var acurso = arrayListOf<String>()
    var aid = arrayListOf<String>()
    var aidAsignatura = arrayListOf<String>()

    adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
anombre)

    database.child("Asignaturas").addValueEventListener(object :
ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            if (snapshot.exists()) {
                for (postSnapshot in snapshot.children) {

```

```

        var idAsignatura = postSnapshot.key

        database.child("Asignaturas-
Alumnos").child(idAsignatura.toString()).child(dni).addValueEventListener(object : ValueEventListener {
            override fun onDataChange(tsnapshot:
DataSnapshot) {

                if (tsnapshot.exists()) {

                    var nombre =
postSnapshot.child("nombre").value
                    var curso =
postSnapshot.child("curso").value
                    var asignatura =
database.child("Asignaturas-Alumnos").child(idAsignatura.toString()).key

                    aidAsignatura.add(asignatura.toString())
                    anombre.add(nombre.toString())
                    acurso.add(curso.toString())
                    listView.adapter = adapter
                }

            }

            override fun onCancelled(error:
DatabaseError) {

                showAlert("Error al acceder a la base de
datos ", ":")

            }

        })

    }

    override fun onCancelled(error: DatabaseError) {
        showAlert("Error al acceder a la base de datos ", ":")
    }

    listView.setOnItemClickListener { parent, view, position, id ->
        pos = position

        showAsignaturaAlumno(aidAsignatura[pos], dni)
    }

}

if(tipo == "cuestionariosAlumno") {
    buttonSeleccionar.visibility = View.INVISIBLE
    var idA= idAsignatura
    var idni = dni

```

```

var atitulo = arrayListOf<String>()
var anotas = arrayListOf<String>()

adapter = ArrayAdapter<this, android.R.layout.simple_list_item_1,
atitulo>
textViewLista.text = "Lista de cuestionarios: "
var aidCuestionarios = arrayListOf<String>()

database.child("Asignaturas-
Cuestionarios").child(idAsignatura).addValueEventListener(object :
ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()) {
            for (postSnapshot in snapshot.children) {
                for (idC in postSnapshot.children) {
                    var idCuestionario = idC.value
                    database.child("Alumnos-
Cuestionarios").child(dni)
                        .addValueEventListener(object :
ValueEventListener {
                            override fun onDataChange(snapshot:
DataSnapshot) {
                                if (snapshot.exists()) {
                                    for (postSnapshot in
snapshot.children) {
                                        if
(postSnapshot.key.toString() == idCuestionario.toString()) {
                                            var titulo =
postSnapshot.child("titulo").value
                                            var nota =
postSnapshot.child("nota").value

anotas.add(nota.toString())
atitulo.add(titulo.toString())

listView.adapter
= adapter
                    }
                }
            }
        }
    }
    override fun onCancelled(error:
DatabaseError) {
        showAlert("Error al acceder a la
base de datos ", ":")
    }
})

```

```

        }
    }
}

override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ", ":")
}

}))

listView.setOnItemClickListener { parent, view, position, id ->
    pos = position
    textViewDesplegadoLista.text = "Titulo: " + atitulo[position]
+
    System.getProperty("line.separator") +
    "Nota: " + anotas[position]
}

}

}

}

private fun cerrarCuestionario(idQuiz: String, idAsignatura: String) {
    database.child("Cuestionarios").child(idQuiz).child("codigo").removeValue()
    onBackPressed()
}

private fun showSeleccionarPregunta(tipo:String,idPregunta:
String,idAsignatura: String) {
    val intent = Intent(this,Seleccionado::class.java).apply {
        putExtra("idPregunta",idPregunta)
        putExtra("tipoSelect",tipo)
        putExtra("idAsignatura",idAsignatura)
    }
    startActivity(intent)
}

private fun matricularAlumno(idAsignatura: String, dni: String) {

    database.child("Asignaturas-
Alumnos").child(idAsignatura).child(dni).child("dni").setValue(dni)//todo
}

private fun añadirCuestionario(idAsignatura: String, idCuestionario:
String) {
    database.child("Asignaturas-
Cuestionarios").child(idAsignatura).child(idCuestionario).setValue(idCuestion
ario)
    showAsignatura(idAsignatura)
}

```



```
    }

    private fun showSeleccionar(tipo: String,idQuiz: String,idAsignatura:
String) {
        val intent = Intent(this,Seleccionado::class.java).apply {
            putExtra("idQuiz",idQuiz)
            putExtra("tipoSelect",tipo)
            putExtra("idAsignatura",idAsignatura)
        }
        startActivity(intent)
    }

    private fun showAsignatura(idAsignatura: String) {
        val intent = Intent(this,AsignaturaProfesor::class.java).apply {
            putExtra("idAsignatura",idAsignatura)
        }
        startActivity(intent)
    }

    private fun showAsignaturaAlumno(idAsignatura: String, dni: String) {
        val intent = Intent(this,AsignaturaAlumno::class.java).apply {
            putExtra("idAsignatura",idAsignatura)
            putExtra("dni",dni)
        }
        startActivity(intent)
    }

    fun showAlert(mensaje:String,value: String){
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Error")
        builder.setMessage("$mensaje" + "$value")
        builder.setNegativeButton("Aceptar" , DialogInterface.OnClickListener
{ dialog, which -> dialog.cancel() })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun showFormQuiz(idQuiz: String, idAsignatura: String) {
        val homeIntent = Intent(this,FormQuiz::class.java).apply {
            putExtra("idQuiz",idQuiz)
            putExtra("idAsignatura",idAsignatura)
            //putExtra("idQuestion",idQuestion)
        }
        startActivity(homeIntent)
    }

    private fun showListaNotas(tipo: String,idAsignatura: String,dni:String)
{
        val homeIntent = Intent(this,Lista::class.java).apply {
            putExtra("ListaDe",tipo)
            putExtra("idAsignatura",idAsignatura)
            putExtra("idAlumno",dni)
        }
        startActivity(homeIntent)
    }
}
```

```

    private fun showListaNotasAsignatura(tipo: String, idAsignatura: String,
idCuestionario: String) {
        val homeIntent = Intent(this,Lista::class.java).apply {
            putExtra("ListaDe",tipo)
            putExtra("idAsignatura",idAsignatura)
            putExtra("idQuiz",idCuestionario)
        }
        startActivity(homeIntent)
    }

    override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {

        if (tipoLista=="asignaturasProfesor") {
            showHome()
        }
        return super.onKeyDown(keyCode, event)
    }

    private fun showNotas(idAsignatura: String, dni :String) {
        val mIntent = Intent(this, Notas::class.java).apply {
            putExtra("tipo","alumno")
            putExtra("idAsignatura", idAsignatura)
            putExtra("dni",dni)
        }
        startActivity(mIntent)
    }

    private fun showHome() {
        val homeIntent = Intent(this,HomeActivity::class.java).apply {
        }
        startActivity(homeIntent)
    }

    private fun showHomeAlumno(dni:String) {
        val homeIntent = Intent(this,HomeAlumno::class.java).apply {
            putExtra("dni",dni)
        }
        startActivity(homeIntent)
    }
}

```

2.12 Seleccionado.kt

```

package com.example.quizu

import android.content.DialogInterface
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.AdapterView
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener

```

```

import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_seleccionado.*
import java.util.*

class Seleccionado : AppCompatActivity() {

    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference
    override fun onCreate(savedInstanceState: Bundle?) {
        val bundle = intent.extras
        val tipo = bundle?.getString("tipoSelect")
        val idQuiz = bundle?.getString("idQuiz")
        val idPregunta = bundle?.getString("idPregunta")
        val idAsignatura = bundle?.getString("idAsignatura")
        var dni = bundle?.getString("dni")
        var mark = bundle?.getDouble("nota")

        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_seleccionado)

        setup(idQuiz ?: "", idPregunta?:"", tipo ?: "", idAsignatura ?:"",
dni ?:"", mark?:0.0)
    }

    private fun setup(

        idQuiz: String,
        idPregunta: String,
        tipo: String,
        idAsignatura: String,
        dni: String,
        mark: Double
    ) {

        buttonLanzarCuestionario.visibility = View.INVISIBLE

        if (tipo=="preguntas") {

            database.child("Preguntas").child(idPregunta).addValueEventListener(object
: ValueEventListener{
                override fun onDataChange(snapshot: DataSnapshot) {
                    var question =
snapshot.child("enunciado").value.toString()
                    var aopcion1 =
snapshot.child("respuesta1").value.toString()
                    var aopcion2 =
snapshot.child("respuesta2").value.toString()
                    var aopcion3 =
snapshot.child("respuesta3").value.toString()
                    var aopcion4 =
snapshot.child("respuesta4").value.toString()

```

```

        var correcta =
snapshot.child("correcta").value.toString()

        when {
            aopcion4 != "null" -> {
                textViewSeleccionadoArriba.text = "Enunciado: " +
question +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1 +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2 +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3 +
                System.getProperty("line.separator") +
                "Opcion 4: " + aopcion4 +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + correcta +
                System.getProperty("line.separator")
            }
            aopcion3 != "null" -> {
                textViewSeleccionadoArriba.text = "Enunciado: " +
question +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1 +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2 +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3 +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + correcta +
                System.getProperty("line.separator")
            }
            else -> {
                textViewSeleccionadoArriba.text = "Enunciado: " +
question +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1 +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2 +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + correcta +
                System.getProperty("line.separator")
            }
        }
    }

    override fun onCancelled(error: DatabaseError) {

    }

})

buttonEliminarSeleccionado.setOnClickListener {
    if (idQuiz != null) {
        controlEliminarPregunta(idQuiz, idPregunta, idAsignatura)
    }
}

```

```

    }

    buttonAtrasSeleccionado.setOnClickListener {
        if (idQuiz != null) {
            onBackPressed()
        }
    }

}

if (tipo=="preguntasBanco") {

database.child("Preguntas").child(idPregunta).addValueEventListener(object
: ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        var question =
snapshot.child("enunciado").value.toString()
        var aopcion1 =
snapshot.child("respuesta1").value.toString()
        var aopcion2 =
snapshot.child("respuesta2").value.toString()
        var aopcion3 =
snapshot.child("respuesta3").value.toString()
        var aopcion4 =
snapshot.child("respuesta4").value.toString()
        var acorrecta =
snapshot.child("correcta").value.toString()

        when {
            aopcion4 != "null" -> {
                textViewSeleccionadoArriba.text = "Enunciado: " +
question +

                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1 +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2 +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3 +
                System.getProperty("line.separator") +
                "Opcion 4: " + aopcion4 +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + acorrecta +
                System.getProperty("line.separator")
            }
            aopcion3 != "null" -> {
                textViewSeleccionadoArriba.text = "Enunciado: " +
question +

                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1 +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2 +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3 +
                System.getProperty("line.separator") +
                "Opcion 4: " + aopcion4 +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + acorrecta +
                System.getProperty("line.separator")
            }
        }
    }
})
}

```

```

        "Correcta: Opcion " + acorrecta +
        System.getProperty("line.separator")
    }
    else -> {
        textViewSeleccionadoArriba.text = "Enunciado: " +
question +
        System.getProperty("line.separator") +
        "Opcion 1: " + aopcion1 +
        System.getProperty("line.separator") +
        "Opcion 2: " + aopcion2+
        System.getProperty("line.separator") +
        "Correcta: Opcion " + acorrecta +
        System.getProperty("line.separator")
    }
}

}

override fun onCancelled(error: DatabaseError) {

}

})

buttonEliminarSeleccionado.setOnClickListener {
    if (idQuiz != null) {
        controlEliminardeBanco(idPregunta,idAsignatura)
    }
}

buttonAtrasSeleccionado.setOnClickListener {
    if (idQuiz != null) {
        onBackPressed()
    }
}

}

if (tipo=="cuestionarios"){

    buttonLanzarCuestionario.visibility = View.VISIBLE
    var question = arrayListOf<String>()
    var numRes = arrayListOf<String>()
    var aopcion1 = arrayListOf<String>()
    var aopcion2 = arrayListOf<String>()
    var aopcion3 = arrayListOf<String>()
    var aopcion4 = arrayListOf<String>()
    var acorrecta = arrayListOf<String>()
    var aIdpregunta = arrayListOf<String>()
    var adapter =
ArrayAdapter<this,android.R.layout.simple_list_item_1,question>

    database.child("Cuestionarios").child(idQuiz).addListenerForSingleValueEvent(
    object : ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {

```

```

        if (snapshot.exists()){
            if(snapshot.child("codigo").exists()) {
                textViewSeleccionadoArriba.text =
                    "Titulo:" +
snapshot.child("Titulo").value.toString() +
                    System.getProperty("line.separator") +
                    "Penalizacion: "+
snapshot.child("Penalizacion").value.toString()+
                    System.getProperty("line.separator") +
                    "Codigo:
"+snapshot.child("codigo").value.toString()
            }else{
                textViewSeleccionadoArriba.text =
                    "Titulo:" +
snapshot.child("Titulo").value.toString() +
                    System.getProperty("line.separator")
+
                    "Penalizacion: "+
snapshot.child("Penalizacion").value.toString()
            }
        }
    }

    override fun onCancelled(error: DatabaseError) {
        TODO("Not yet implemented")
    }

})

database.child("Cuestionarios").child(idQuiz).child("Preguntas").addValueEventListener(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()) {
            for(postSnapshot in snapshot.children ){
                var enunciado =
postSnapshot.child("enunciado").value
                var numeroRespuestas =
postSnapshot.child("numeroRespuestas").value
                var opcion1 =
postSnapshot.child("respuesta1").value
                var opcion2 =
postSnapshot.child("respuesta2").value
                var opcion3 =
postSnapshot.child("respuesta3").value
                var opcion4
=postSnapshot.child("respuesta4").value
                var correcta =
postSnapshot.child("correcta").value
                var idPregunta = postSnapshot.key

                numRes.add(numeroRespuestas.toString())
                aopcion1.add(opcion1.toString())
                aopcion2.add(opcion2.toString())
                aopcion3.add(opcion3.toString())
                aopcion4.add(opcion4.toString())
                acorrecta.add(correcta.toString())
            }
        }
    }
})

```

```

        question.add(enunciado.toString())
        aIdpregunta.add(idPregunta.toString())

        listViewSeleccionado.adapter = adapter
    }
}
}
override fun onCancelled(error: DatabaseError) {
    showAlert("Error al acceder a la base de datos ", ":")
}
})

listViewSeleccionado.setOnItemClickListener { parent, view,
position, id ->
    when {
        aopcion4[position] != "null" -> {
            textViewSelccionadoAbajo.text = "Enunciado: " +
question[position] +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1[position] +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2[position] +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3[position] +
                System.getProperty("line.separator") +
                "Opcion 4: " + aopcion4[position] +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + acorrecta[position] +
                System.getProperty("line.separator")
        }
        aopcion3[position] != "null" -> {
            textViewSelccionadoAbajo.text = "Enunciado: " +
question[position] +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1[position] +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2[position] +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3[position] +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + acorrecta[position] +
                System.getProperty("line.separator")
        }
        else -> {
            textViewSelccionadoAbajo.text = "Enunciado: " +
question[position] +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1[position] +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2[position] +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + acorrecta[position] +
                System.getProperty("line.separator")
        }
    }
}
}

```



```

    }

    buttonEliminarSeleccionado.setOnClickListener {
        if (idQuiz != null) {
            controlEliminarCuestionario(idQuiz, idAsignatura)
        }
    }

    buttonAtrasSeleccionado.setOnClickListener {
        onBackPressed()
    }

    buttonLanzarCuestionario.setOnClickListener {
        var codigo = lanzarCuestionario(idQuiz)
        showCodigo(codigo, idAsignatura)
    }

}
if (tipo=="nota") {

database.child("Cuestionarios").child(idQuiz).addValueEventListener(object :
ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        textViewSeleccionadoArriba.text = "DNI: " + dni+
            System.getProperty("line.separator") +
            "Cuestionario: " +
snapshot.child("Titulo").value.toString()+
            System.getProperty("line.separator") +
            "Nota: " + mark.toString() +
            System.getProperty("line.separator")
    }

    override fun onCancelled(error: DatabaseError) {
        TODO("Not yet implemented")
    }
})

buttonLanzarCuestionario.visibility = View.INVISIBLE
buttonEliminarSeleccionado.visibility = View.INVISIBLE
var question = arrayListOf<String>()
var numRes = arrayListOf<String>()
var aopcion1 = arrayListOf<String>()
var aopcion2 = arrayListOf<String>()
var aopcion3 = arrayListOf<String>()
var aopcion4 = arrayListOf<String>()
var acorrecta = arrayListOf<String>()
var aIdpregunta = arrayListOf<String>()
var adapter =
ArrayAdapter<this, android.R.layout.simple_list_item_1, question>

```

```

database.child("Cuestionarios").child(idQuiz).child("Preguntas").addValueEventListener(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()) {
            for(postSnapshot in snapshot.children){
                var enunciado =
                postSnapshot.child("enunciado").value
                var numeroRespuestas =
                postSnapshot.child("numeroRespuestas").value
                var opcion1 =
                postSnapshot.child("respuesta1").value
                var opcion2 =
                postSnapshot.child("respuesta2").value
                var opcion3 =
                postSnapshot.child("respuesta3").value
                var opcion4
                =postSnapshot.child("respuesta4").value
                var correcta =
                postSnapshot.child("correcta").value
                var idPregunta = postSnapshot.key

                numRes.add(numeroRespuestas.toString())
                aopcion1.add(opcion1.toString())
                aopcion2.add(opcion2.toString())
                aopcion3.add(opcion3.toString())
                aopcion4.add(opcion4.toString())
                acorrecta.add(correcta.toString())
                question.add(enunciado.toString())
                aIdpregunta.add(idPregunta.toString())

                listViewSeleccionado.adapter = adapter
            }
        }
    }
    override fun onCancelled(error: DatabaseError) {
        showAlert("Error al acceder a la base de datos ", ":")
    }
})

listViewSeleccionado.setOnItemClickListener { parent, view,
position, id ->
    when {
        aopcion4[position] != "null" -> {
            textViewSelccionadoAbajo.text = "Enunciado: " +
question[position] +
                System.getProperty("line.separator") +
                "Opcion 1: " + aopcion1[position] +
                System.getProperty("line.separator") +
                "Opcion 2: " + aopcion2[position] +
                System.getProperty("line.separator") +
                "Opcion 3: " + aopcion3[position] +
                System.getProperty("line.separator") +
                "Opcion 4: " + aopcion4[position] +
                System.getProperty("line.separator") +
                "Correcta: Opcion " + acorrecta[position] +
                System.getProperty("line.separator")
        }
    }
}

```

```

    }
    aopcion3[position] != "null" -> {
        textViewSelccionadoAbajo.text = "Enunciado: " +
question[position] +
        System.getProperty("line.separator") +
        "Opcion 1: " + aopcion1[position] +
        System.getProperty("line.separator") +
        "Opcion 2: " + aopcion2[position] +
        System.getProperty("line.separator") +
        "Opcion 3: " + aopcion3[position] +
        System.getProperty("line.separator") +
        "Correcta: Opcion " + acorrecta[position] +
        System.getProperty("line.separator")
    }
    else -> {
        textViewSelccionadoAbajo.text = "Enunciado: " +
question[position] +
        System.getProperty("line.separator") +
        "Opcion 1: " + aopcion1[position] +
        System.getProperty("line.separator") +
        "Opcion 2: " + aopcion2[position] +
        System.getProperty("line.separator") +
        "Correcta: Opcion " + acorrecta[position] +
        System.getProperty("line.separator")
    }
    }
    }

    buttonAtrasSeleccionado.setOnClickListener {

        showHomeAlumno(dni)

    }

}
}

```

```

private fun lanzarCuestionario(idQuiz: String): String{
    var codigo = valorRandom(10000..99999)

    database.child("Cuestionarios").child(idQuiz).child("codigo").setValue(codigo
)
    return codigo.toString()

}

```

```

fun valorRandom(valores: IntRange) : Int {
    var r = Random()
    var valorRandom = r.nextInt(valores.last - valores.first) +
valores.first
    return valorRandom
}

```

```

    }

    private fun controlEliminarCuestionario(idQuiz:
String,idAsignatura:String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
        builder.setMessage("¿Esta seguro de eliminar el cuestionario?")
        builder.setNegativeButton("Cancelar", DialogInterface.OnClickListener { dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Eliminar", DialogInterface.OnClickListener { dialog, which ->
            var key = idQuiz
            database.child("Cuestionarios").child(key).removeValue()
            database.child("Asignaturas-
Cuestionarios").child(idAsignatura).child(key).removeValue()
            showListCuestionarios("cuestionariosProfesor",idAsignatura)
        })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun showListCuestionarios(tipo: String, idAsignatura: String) {
        val homeIntent = Intent(this,Lista::class.java).apply {
            putExtra("ListaDe",tipo)
            putExtra("idAsignatura",idAsignatura)
        }
        startActivity(homeIntent)
    }

    private fun controlEliminarPregunta(idQuiz:
String,idPregunta:String,idAsignatura: String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
        builder.setMessage("¿Esta seguro de eliminar la pregunta del
cuestionario?")
        builder.setNegativeButton("Volver al cuestionario",
DialogInterface.OnClickListener { dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Eliminar", DialogInterface.OnClickListener { dialog, which ->
            borrarPregunta(idQuiz,idPregunta)
            showFormQuiz(idQuiz,idAsignatura)
        })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun controlEliminardeBanco(idPregunta: String,idAsignatura:
String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
    }

```

```

        builder.setMessage("¿Quiere eliminarla del banco de preguntas?")
        builder.setNegativeButton("No", DialogInterface.OnClickListener {
dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Si", DialogInterface.OnClickListener {
dialog, which ->
            borrarPreguntaBanco(idPregunta,idAsignatura)
            showAsignaturaProfesor(idAsignatura)
        })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun borrarPreguntaBanco(idPregunta: String,idAsignatura: String)
    {
        var key = idPregunta
        database.child("Preguntas").child(key).removeValue()
        database.child("Asignaturas-
Preguntas").child(idAsignatura).child(key).removeValue()
    }

    private fun borrarPregunta(idQuiz: String, idPregunta: String) {
        var keyQ = idQuiz
        var keyP = idPregunta

        database.child("Cuestionarios").child(keyQ).child("Preguntas").child(keyP).re
moveValue()
    }

    private fun showFormQuiz(idQuiz:String,idAsignatura: String) {
        val homeIntent = Intent(this,FormQuiz::class.java).apply {
            putExtra("idQuiz",idQuiz)
            putExtra("idAsignatura",idAsignatura)
        }
        startActivity(homeIntent)
    }

    private fun showHomeAlumno(dni: String) {
        val homeIntent = Intent(this,HomeAlumno::class.java).apply {
            putExtra("dni",dni)
        }
        startActivity(homeIntent)
    }

    private fun showFormQuestion(idQuiz: String?, idPregunta: String) {
        val homeIntent = Intent(this,FormQuestion::class.java).apply {
            putExtra("idQuiz",idQuiz)
            putExtra("iPregunta",idPregunta)
        }
        startActivity(homeIntent)
    }

    fun showAlert(mensaje:String,value: String){
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Error")
        builder.setMessage("$mensaje" + "$value")
        builder.setNegativeButton("Aceptar" , DialogInterface.OnClickListener

```

```

{ dialog, which -> dialog.cancel() })
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

private fun showAsignaturaProfesor(idAsignatura: String) {
    val homeIntent = Intent(this, AsignaturaProfesor::class.java).apply {
        putExtra("idAsignatura", idAsignatura)
    }
    startActivity(homeIntent)
}
private fun showCodigo(codigo: String, idAsignatura: String) {
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Cuestionario activado")
    builder.setMessage("Codigo:" + "$codigo")
    builder.setPositiveButton("Aceptar", DialogInterface.OnClickListener {
        dialog, which -> showAsignaturaProfesor(idAsignatura)})
    val dialog: AlertDialog = builder.create()
    dialog.show()
}
}
}

```

2.13 SolveQuiz.kt

```

package com.example.quizu

import android.content.DialogInterface
import android.content.Intent
import android.graphics.Color
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.KeyEvent
import android.view.View
import androidx.appcompat.app.AlertDialog
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_solve_quiz.*
import com.example.quizu.Pregunta as Question

class SolveQuiz : AppCompatActivity() {
    val database = Firebase.database("https://quizu-f4483-default-
rtdb.europe-west1.firebaseio.com/").reference

    private var currentPos: Int = 1
    private var questionList: ArrayList<Question> = ArrayList()
    private var correctAnswer: Int = 0
    private var mark = -1.0f
    private var penalty = 33
    var preguntas = HashMap<Int, Int>()
    var control1 = 0.0

```

```

var control2 = 0.0
var control3= 0.0
var control4=0.0

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_solve_quiz)

    val bundle = intent.extras
    val dni = bundle?.getString("dni")
    val idQuiz = bundle?.getString("idQuiz")

    if (idQuiz != null) {

database.child("Cuestionarios").child(idQuiz).addValueEventListener(object
: ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {
            textViewTituloQuiz.text =
snapshot.child("Titulo").value.toString()

                penalty =
snapshot.child("Penalizacion").value.toString().toInt()

        }

        override fun onCancelled(error: DatabaseError) {

        }

    })
    }

    setup(dni ?: "", idQuiz ?: "")

}

private fun setupRevision(dni: String, idQuiz: String) {
    val color = textViewEnunciadoSolve.currentTextColor
    val nota = "%.2f".format(mark)
    textViewTituloQuiz.text = "Revision: Su nota es de "+ nota

    radioButtonOpcion1.setOnClickListener{

        //radioGroupQuiz.clearCheck()

    }

    radioButtonOpcion2.setOnClickListener{

        //pQuiz.clearCheck()

```

```

    }

    radioButtonOpcion3.setOnClickListener{

        // radioButtonQuiz.clearCheck()

    }

    radioButtonOpcion4.setOnClickListener{

        //radioGroupQuiz.clearCheck()

    }

    when {
        currentPos <= questionList.size -> {
            setQuestion()

            revision(checkAnswer(),color)

            buttonPreguntaAnterior.setOnClickListener {

                setQuestion()

                currentPos -= 1
                setQuestion()
                revision(checkAnswer(),color)

            }

            buttonAvanzar.setOnClickListener {

                if(currentPos<questionList!!.size) {

                    currentPos ++
                    setQuestion()
                    revision(checkAnswer(),color)

                }else{

                    revision(checkAnswer(), color)
                    controlfinalizarRevision(dni)

                }

            }

        }else ->{

```



```

//No debería de entrar nunca, si lo hace volver a la
primera pregunta
//Control de errores

mark = 0.0F
currentPos = 1
//setQuestion()
}
}

buttonCancelar.setOnClickListener {
    controlfinalizarRevision(dni)
}

}

fun setup(dni:String,idQuiz:String) {

    if (idQuiz != null) {

database.child("Cuestionarios").child(idQuiz).child("Preguntas").addValueEven
tListener(object :ValueEventListener{
    override fun onDataChange(snapshot: DataSnapshot) {
        if (snapshot.exists()) {
            for (postSnapshot in snapshot.children) {
                val idPregunta = postSnapshot.key
                val numeroRespuestas =
postSnapshot.child("numeroRespuestas").value
                val enunciado =
postSnapshot.child("enunciado").value
                val respuesta1 =
postSnapshot.child("respuesta1").value
                val respuesta2 =
postSnapshot.child("respuesta2").value
                val respuesta3 =
postSnapshot.child("respuesta3").value
                val respuesta4 =
postSnapshot.child("respuesta4").value
                val correcta =
postSnapshot.child("correcta").value.toString().toInt()
                val questionDetails =
Question(idPregunta.toString(),enunciado.toString(),numeroRespuestas.toString
(),
                ,respuesta1.toString(),
respuesta2.toString(),respuesta3.toString()
                ,respuesta4.toString(), correcta)
                questionList.add(questionDetails)

            }

            var correcta = HashMap<Int, Int>()

```

```

radioButtonOpcion1.setOnClickListener{
    control2 = 0.0
    control3= 0.0
    control4=0.0
    control1++
    if (control1%2 ==0.0) {
        radioGroupQuiz.clearCheck()
        control1 = 0.0
    }
}

radioButtonOpcion2.setOnClickListener{

    control1 = 0.0
    control3 = 0.0
    control4 = 0.0
    control2++
    if (control2 % 2 == 0.0) {
        radioGroupQuiz.clearCheck()
        control2 = 0.0
    }
}

radioButtonOpcion3.setOnClickListener{

    control1 = 0.0
    control2 = 0.0
    control4 = 0.0
    control3++
    if (control3 % 2 == 0.0) {
        radioGroupQuiz.clearCheck()
        control3 = 0.0
    }
}

radioButtonOpcion4.setOnClickListener{

    control4++
    control1 = 0.0
    control2 = 0.0
    control3 = 0.0

    if (control4 % 2 == 0.0) {
        radioGroupQuiz.clearCheck()
        control4 = 0.0
    }
}

when {
    currentPos <= questionList.size -> {
        setQuestion()
    }
}

```

```

buttonPreguntaAnterior.setOnClickListener {

    control1 = 0.0
    control2 = 0.0
    control3 = 0.0
    control4 = 0.0

    preguntas.put(currentPos,
correctAnswer)

    correcta.put(currentPos,
correctAnswer)

    currentPos -= 1
    setQuestion()
    saveQuestion(preguntas)

}

buttonAvanzar.setOnClickListener {

    control1 = 0.0
    control2 = 0.0
    control3 = 0.0
    control4 = 0.0

    if(currentPos<questionList!!.size) {

        preguntas.put(currentPos,
correctAnswer)

        correcta.put(currentPos,
correctAnswer)

        currentPos++
        setQuestion()
        saveQuestion(preguntas)

    }else{

        preguntas.put(currentPos,
correctAnswer)

        correcta.put(currentPos,
correctAnswer)

        mark(preguntas, correcta)
        saveQuestion(preguntas)

        if (mark < 0.0)
            mark = 0.0F

        controlfinalizar(dni, idQuiz)
    }
}

```

```

        }
    }

    }else ->{
        //No debería de entrar nunca, si lo hace volver
        //Control de errores

        mark = 0.0F
        currentPos = 1

    }
}

buttonCancelar.setOnClickListener {

    controlExit()

}

}

}

override fun onCancelled(error: DatabaseError) {

}

})

}

}

```

a la primera pregunta

```

private fun revision(checkAnswer: Int, color: Int) {
    val question = questionList[currentPos -1]

    when(preguntas[currentPos]){
        1->{
            radioButtonOpcion1.setTextColor(Color.parseColor("#FF0000"))
            radioButtonOpcion2.setTextColor(color)
            radioButtonOpcion3.setTextColor(color)
            radioButtonOpcion4.setTextColor(color)

        }
        2->{
            radioButtonOpcion2.setTextColor(Color.parseColor("#FF0000"))
            radioButtonOpcion1.setTextColor(color)
            radioButtonOpcion3.setTextColor(color)
            radioButtonOpcion4.setTextColor(color)

        }
        3->{

```

```

        radioButtonOpcion3.setTextColor(Color.parseColor("#FF0000"))
        radioButtonOpcion2.setTextColor(color)
        radioButtonOpcion1.setTextColor(color)
        radioButtonOpcion4.setTextColor(color)
    }
    4->{
        radioButtonOpcion4.setTextColor(Color.parseColor("#FF0000"))
        radioButtonOpcion2.setTextColor(color)
        radioButtonOpcion1.setTextColor(color)
        radioButtonOpcion4.setTextColor(color)
    }
}
when(question.correcta){
    1->{
        radioButtonOpcion1.setTextColor(Color.parseColor("#00FF00"))

    }
    2->radioButtonOpcion2.setTextColor(Color.parseColor("#00FF00"))
    3->radioButtonOpcion3.setTextColor(Color.parseColor("#00FF00"))
    4->radioButtonOpcion4.setTextColor(Color.parseColor("#00FF00"))
}
}

private fun guardaNota(dni: String, idQuiz: String, nota: Float) {

    database.child("Alumnos").child(dni).addValueEventListener(object
:ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {
            database.child("Alumnos-
Cuestionarios").child(dni).child("nombre").setValue(snapshot.child("nombre").
value.toString())
        }
        override fun onCancelled(error: DatabaseError) {
        }
    })

    database.child("Cuestionarios").child(idQuiz).addValueEventListener(object
:ValueEventListener{
        override fun onDataChange(snapshot: DataSnapshot) {
            database.child("Alumnos-
Cuestionarios").child(dni).child(idQuiz).child("titulo").setValue(snapshot.ch
ild("Titulo").value.toString())
        }
        override fun onCancelled(error: DatabaseError) {
        }
    })
    database.child("Alumnos-
Cuestionarios").child(dni).child(idQuiz).child("nota").setValue(nota)
}

private fun controlExit() {
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Cuidado!")
    builder.setMessage("No has completado el cuestionario, ¿seguro que
quieres cancelar?")
}

```

```

        builder.setNegativeButton("Volver", DialogInterface.OnClickListener {
dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Salir", DialogInterface.OnClickListener {
dialog, which -> onBackPressed() })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun controlExitRevision() {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
        builder.setMessage("No has completado la revision, ¿seguro que
quieres cancelar?")
        builder.setNegativeButton("Volver", DialogInterface.OnClickListener {
dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Salir", DialogInterface.OnClickListener {
dialog, which -> onBackPressed() })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun controlfinalizar(dni: String, idQuiz: String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
        builder.setMessage("Una vez enviado no se puede modificar, ¿Deseas
terminar el cuestionario?")
        builder.setNegativeButton("Volver", DialogInterface.OnClickListener {
dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Enviar", DialogInterface.OnClickListener {
dialog, which ->
            guardaNota(dni,idQuiz,mark)
            currentPos=1
            showNota(mark)
            setupRevision(dni,idQuiz)
            //setup(dni,idQuiz,mark)
        })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun controlfinalizarRevision(dni: String) {
        val builder = AlertDialog.Builder(this)
        builder.setTitle("Revision Completada")
        builder.setMessage("Una vez salgas no se puede volver, ¿Deseas
terminar la revision?")
        builder.setNegativeButton("Volver", DialogInterface.OnClickListener {
dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Salir", DialogInterface.OnClickListener {
dialog, which ->
            showHome(dni)
        })
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    private fun saveQuestion(preguntas: HashMap<Int, Int>) {

```

```

var checked = 0
if(preguntas.get(currentPos) == null){
    preguntas.put(currentPos,0)
}else{
    checked = preguntas.get(currentPos)!!
    when(checked){
        1 ->{
            radioButtonOpcion1.isChecked = true
            control1++
        }
        2 ->{
            radioButtonOpcion2.isChecked = true
            control2++
        }
        3 ->{
            radioButtonOpcion3.isChecked = true
            control3++
        }
        4-> {
            radioButtonOpcion4.isChecked = true
            control4++
        }
        else ->{
            radioGroupQuiz.clearCheck()
        }
    }
}
}

private fun checkAnswer():Int{

    var check = 0
    if (radioButtonOpcion1.isChecked)
        check = 1

    if (radioButtonOpcion2.isChecked)
        check = 2

    if (radioButtonOpcion3.isChecked)
        check = 3

    if (radioButtonOpcion4.isChecked)
        check = 4

    radioGroupQuiz.clearCheck()
    return check

}

private fun mark(preguntas: HashMap<Int, Int>, correcta: HashMap<Int,
Int>){
    var resultado = 0.0
    var mal = penalty
    var i = 1

```

```

var valor = 10.0/preguntas.size
var incorret = valor * (mal/100.0)

while (i<=preguntas.size) {

    when (preguntas.get(i)) {

        correcta.get(i) -> {
            resultado = resultado + valor

        }

        0 -> {

            resultado = resultado

        }

        else -> {

            resultado = resultado - incorret

        }

    }
    i++
}
mark = resultado.toFloat()
}

private fun setQuestion(){

    val question = questionList[currentPos -1]

    when(question.numeroRespuestas){
        "4"->{

            textViewNumeroPregunta.text = "Pregunta " + "$currentPos"
            textViewEnunciadoSolve.text = question.enunciado
            radioButtonOpcion1.text = question.opcion1
            radioButtonOpcion2.text = question.opcion2
            radioButtonOpcion3.text = question.opcion3
            radioButtonOpcion4.text = question.opcion4
            correctAnswer = question.correcta

        }
        "3"->{
            textViewNumeroPregunta.text = "Pregunta " + "$currentPos"
            textViewEnunciadoSolve.text = question.enunciado
            radioButtonOpcion1.text = question.opcion1
            radioButtonOpcion2.text = question.opcion2
            radioButtonOpcion3.text = question.opcion3
            radioButtonOpcion4.visibility =View.INVISIBLE
            correctAnswer = question.correcta
        }
    }
}

```



```

    }else->{
        textViewNumeroPregunta.text = "Pregunta " + "$currentPos"
        textViewEnunciadoSolve.text = question.enunciado
        radioButtonOpcion1.text = question.opcion1
        radioButtonOpcion2.text = question.opcion2
        radioButtonOpcion3.visibility = View.INVISIBLE
        radioButtonOpcion4.visibility = View.INVISIBLE
        correctAnswer = question.correcta
    }
}

if(currentPos==1) {
    buttonPreguntaAnterior.visibility = View.INVISIBLE
}else{
    buttonPreguntaAnterior.visibility = View.VISIBLE
}
//Se comprueba si es la ultima pregunta si lo es se cambia el mensaje
del boton
if (currentPos == questionList!!.size) {
    buttonAvanzar.text = "Finalizar"
}else{
    buttonAvanzar.text = "Siguiente"
}
}

private fun showNota(tipo: String, dni: String,idQuiz: String,nota:
Double) {
    val mIntent = Intent(this,Seleccionado::class.java).apply {
        putExtra("idQuiz",idQuiz)
        putExtra("tipoSelect",tipo)
        putExtra("dni",dni)
        putExtra("nota",nota)
    }
    startActivity(mIntent)
}

fun showNota(nota: Float){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Cuestionario Finalizado")
    builder.setMessage("Nota final: "+$nota")
    builder.setPositiveButton("Iniciar Revision",null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

private fun showHome(dni: String) {
    val mIntent = Intent(this,HomeAlumno::class.java).apply {
        putExtra("dni",dni)
    }
    startActivity(mIntent)
}

fun showAlert(mensaje:String){
    val builder = AlertDialog.Builder(this)

```

```

        builder.setTitle("Error")
        builder.setMessage(mensaje)
        builder.setPositiveButton("Aceptar",null)
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {

        val builder = AlertDialog.Builder(this)
        builder.setTitle("Cuidado!")
        builder.setMessage("No has completado el cuestionario, ¿seguro que  
quieres cancelar?")
        builder.setNegativeButton("Volver", DialogInterface.OnClickListener {
            dialog, which -> dialog.cancel() })
        builder.setPositiveButton("Salir", DialogInterface.OnClickListener {
            dialog, which -> onBackPressed() })
        val dialog: AlertDialog = builder.create()
        dialog.show()
        return false
    }
}

```

2.14 Notas.kt

```

package com.example.quizu

import android.Manifest.permission.READ_EXTERNAL_STORAGE
import android.Manifest.permission.WRITE_EXTERNAL_STORAGE
import android.content.Context
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.core.content.FileProvider
import com.github.mikephil.charting.data.BarData
import com.github.mikephil.charting.data.BarDataSet
import com.github.mikephil.charting.data.BarEntry
import com.github.mikephil.charting.utils.ColorTemplate
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase
import kotlinx.android.synthetic.main.activity_notas.*
import java.io.*

class Notas : AppCompatActivity() {
    var adni = arrayListOf<String>()
    var anombre = arrayListOf<String>()

```

```

var aCuestionarios = arrayListOf<String>()
val notas = arrayListOf<Cuestionario>()
var aTitulo = arrayListOf<String>()
val database =
    Firebase.database("https://quizu-f4483-default-rtdb.europe-
west1.firebaseio.com/app/").reference
var control = false
var medias = ArrayList<BarEntry>()
var labelsG = ArrayList<String>()
var participaciones = ArrayList<BarEntry>()
private var notasList: ArrayList<Cuestionario> = ArrayList()

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_notas)

    val bundle = intent.extras
    var idAsignatura = bundle?.getString("idAsignatura")
    var tipo = bundle?.getString("tipo")
    var dni = bundle?.getString("dni")
    var nombre = bundle?.getString("nombre")

    if ((ContextCompat.checkSelfPermission(
        this,
        READ_EXTERNAL_STORAGE
    ) != PackageManager.PERMISSION_GRANTED) ||
        (ContextCompat.checkSelfPermission(this, WRITE_EXTERNAL_STORAGE)
!=
        PackageManager.PERMISSION_GRANTED)
    ) {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(
                READ_EXTERNAL_STORAGE,
                WRITE_EXTERNAL_STORAGE
            ),
            123
        )
    }

    val prefs = getSharedPreferences(getString(R.string.prefs_file),
Context.MODE_PRIVATE)
    val email = prefs.getString("email", null)

    if (idAsignatura != null) {
        database.child("Asignaturas-Cuestionarios").child(idAsignatura)
            .addValueEventListener(object :
                ValueEventListener {
                    override fun onDataChange(snapshot: DataSnapshot) {
                        if (snapshot.exists()) {
                            for (postSnapshot in snapshot.children) {
                                var id = ""
                                if (postSnapshot.exists()) {
                                    id = postSnapshot.value.toString()
                                    aCuestionarios.add(id)
                                }
                            }
                        }
                    }
                })
    }
}

```

```

        database.child("Cuestionarios").child(id)
            .addValueEventListener(object :
                ValueEventListener {
                    override fun onDataChange(snapshot:
DataSnapshot) {

                        if (snapshot.exists()) {
                            var titulo =

snapshot.child("Titulo").value.toString()

                            aTitulo.add(titulo)

                        }
                    }

                    override fun onCancelled(error:
DatabaseError) {

                    }

                })
            })

        override fun onCancelled(error: DatabaseError) {

        }

    })

    if (idAsignatura != null) {
        database.child("Asignaturas-Alumnos").child(idAsignatura)
            .addValueEventListener(object :
                ValueEventListener {
                    override fun onDataChange(snapshot: DataSnapshot) {
                        if (snapshot.exists()) {
                            for (postSnapshot in snapshot.children) {
                                var dni = ""
                                if (postSnapshot.exists()) {
                                    dni =
postSnapshot.child("dni").value.toString()
                                    adni.add(dni)

                                }

                                database.child("Alumnos").child(dni)
                                    .addValueEventListener(object :
                                        ValueEventListener {
                                            override fun onDataChange(ssnapshot:
DataSnapshot) {

                                                if (ssnapshot.exists()) {
                                                    var nombre =

```

```

ssnapshot.child("nombre").value.toString()
                                anombre.add(nombre)
                                }
                                }
                                override fun onCancelled(error:
DatabaseError) {
                                }
                                })
                                }
                                }
                                }
                                override fun onCancelled(error: DatabaseError) {
                                }
                                })
                                }

database.child("Alumnos-Cuestionarios")
    .addListenerForSingleValueEvent(object :
        ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()) {

                    for (postSnapshot in snapshot.children) {
                        var dni = ""
                        if (postSnapshot.exists()) {
                            dni = postSnapshot.key.toString()
                        }
                        database.child("Alumnos-
Cuestionarios").child(dni)
                            .addListenerForSingleValueEvent(
                                object : ValueEventListener {
                                    override fun onDataChange(esnapshot:
DataSnapshot) {

                                        if (esnapshot.exists()) {

                                            var idA = ""
                                            for (apostSnapshot in
esnapshot.children) {

                                                if

                                                if

                                                .exists()
                                                ) {

                                                    var nota =

```

```

apostSnapshot.child("nota").value
apostSnapshot.key.toString()
Cuestionario(
    nota.toString().toFloat(),

    idA =
    var quiz =

        idA,
        dni, ""
    )
    notas.add(quiz)

    notasList.add(quiz)

    }
    }
    }

database.removeEventListener(this);
}

DatabaseError) {
    override fun onCancelled(error:
    }

    })

    }

    }

    override fun onCancelled(error: DatabaseError) {
    }

    })

    if (idAsignatura != null) {
        barchart(idAsignatura)
    }
    setup(idAsignatura ?: "", tipo ?: "", dni ?: "", email ?: "", nombre
?: "")

```

```

    }

    private fun barchart(idAsignatura: String) {

        var idcuestionarios = arrayListOf<String>()
        var cuestionarios = arrayListOf<Cuestionario>()
        database.child("Asignaturas-
Cuestionarios").child(idAsignatura).get().addOnSuccessListener {
            for(snapshot in it.children){
                if (snapshot.exists()) {

                    if(snapshot.exists()){
                        idcuestionarios.add(snapshot.value.toString())
                    }

                }
            }
        }

        database.child("Alumnos-Cuestionarios").get().addOnSuccessListener {
            for(snapshot in it.children){
                if (snapshot.exists()) {
                    for (postsnapshot in snapshot.children) {
                        if (postsnapshot.child("nota").exists()) {
                            var nota = postsnapshot.child("nota").value
                            var idA = postsnapshot.key.toString()
                            var dni = snapshot.key.toString()
                            var titulo = postsnapshot.child("titulo").value

                            for (j in idcuestionarios){
                                if (j == idA){
                                    if (nota != null) {

                                        var quiz = Cuestionario(
                                            nota.toString().toFloat(),
                                            idA,
                                            dni,
                                            titulo.toString()
                                        )
                                        cuestionarios.add(quiz)

                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

database.child("Alumnos-Cuestionarios").get().addOnSuccessListener {
    val entries = ArrayList<BarEntry>()
    val labels = ArrayList<String>()
    val entriesP = ArrayList<BarEntry>()

    var c = ArrayList<String>()
    for (f in idcuestionarios) {
        for(j in cuestionarios){
            if (j.id == f)
                c.add(j.titulo)
        }
    }

    val cuestionariosactivados = ArrayList<String>()

    for (f in c){
        if (!(labels.contains(f))){
            labels.add(f)
            cuestionariosactivados.add(f)
        }
    }

    var i=0
    var contador = 0
    var media = 0.0f
    var presentacion = 0.0f

    for (f in cuestionariosactivados) {
        for(j in cuestionarios){
            if (f == j.titulo) {
                media = media + j.nota
                contador++
            }
        }
        media = media/contador
        entries.add(BarEntry(media, i))
        presentacion = (100.0f /(adn.size/contador))
        entriesP.add(BarEntry(presentacion, i))
        i++
        contador = 0
        media = 0.0f
    }

    medias = entries
    participaciones = entriesP
    labelsG = labels
    val barDataSet = BarDataSet(entries, "Cells")

    val data = BarData(labels, barDataSet)
    barChart.axisRight.setAxisMaxValue(10f)
    barChart.axisLeft.setAxisMaxValue(10f)
    barChart.data = data // set the data and list of lables into
    //barChart.setVisibleYRangeMaximum(10.0f,)

    barChart.setDescription("Notas Cuestionarios") // set the

```

chart

description

```

        barChart.axisRight.setStartAtZero(true)
        barChart.axisLeft.setStartAtZero(true)

        barDataSet.setColors(ColorTemplate.COLORFUL_COLORS)
        //barDataSet.color =
resources.getColor(R.color.browser_actions_bg_grey)

        barChart.legend.calculatedLabelSizes

        barChart.animateY(2000)

    }

}

fun Guardar(texto: String, idAsignatura: String){//, ficheroFisico :
File){
    try {
        val rutaSd = baseContext.getExternalFilesDir(null)?.absolutePath
        val miCarpeta = File(rutaSd, "Notas")
        if (!miCarpeta.exists()) {
            miCarpeta.mkdir()
        }
        val ficheroFisico = File(miCarpeta, "$idAsignatura.csv")
        ficheroFisico.appendText("$texto")
        //ficheroFisico.delete()
    }catch (e: Exception){
        Toast.makeText(
            this,
            "No se ha podido Guardar",
            Toast.LENGTH_LONG
        ).show()
    }
}

fun Cargar(idAsignatura: String): String{
    var texto = ""
    try {

        val rutaSd = baseContext.getExternalFilesDir(null)?.absolutePath
        val miCarpeta = File(rutaSd, "Notas")
        val ficheroFisico = File(miCarpeta, "$idAsignatura.csv")
        val fichero =
BufferedReader(InputStreamReader(FileInputStream(ficheroFisico)))
        texto = fichero.use(BufferedReader::readText)
        Toast.makeText(
            this,
            rutaSd,
            Toast.LENGTH_LONG

```

```

        ).show()
    }catch (e: Exception){
        //textViewArchivo.text = "No existe fichero de notas de esta
        asignatura, pinche el boton de exportar para crearlo"
    }
    return texto
}

fun Eliminar(idAsignatura: String){

    try {

        val rutaSd = baseContext.getExternalFilesDir(null)?.absolutePath
        val miCarpeta = File(rutaSd, "Notas")
        val ficheroFisico = File(miCarpeta, "$idAsignatura.csv")
        ficheroFisico.delete()
    }catch (e: Exception){

    }

}

private fun setup(
    idAsignatura: String,
    tipo: String,
    dni: String,
    email: String,
    nombre: String
) {

    if (tipo == "asignatura") {
        if (idAsignatura != "") {

            buttonGrafica.setOnClickListener {
                if (buttonGrafica.text == "Grafica Participacion"){
                    val barDataSet = BarDataSet(participaciones, "Cells")
                    barChart.axisRight.setAxisMaxValue(100f)
                    barChart.axisLeft.setAxisMaxValue(100f)

                    val data = BarData(labelsG, barDataSet)
                    barChart.data = data // set the data and list of
                    //barChart.setVisibleYRangeMaximum(10.0f,)

                    barChart.setDescription("Participacion
                    Cuestionarios") // set the description

                    barDataSet.setColors(ColorTemplate.COLORFUL_COLORS)
                    //barDataSet.color =
                    resources.getColor(R.color.browser_actions_bg_grey)

                    barChart.legend.calculatedLabelSizes

                    barChart.animateY(2000)
                }
            }
        }
    }
}

```

```

        buttonGrafica.text = "Grafica Medias"

    }else if (buttonGrafica.text == "Grafica Medias"){
        val barDataSet = BarDataSet(medias, "Cells")

        barChart.axisRight.setAxisMaxValue(10f)
        barChart.axisLeft.setAxisMaxValue(10f)
        val data = BarData(labelsG, barDataSet)
        barChart.data = data // set the data and list of
        //barChart.setVisibleYRangeMaximum(10.0f,)

        barChart.setDescription("Notas Cuestionarios") //

        barDataSet.setColors(ColorTemplate.COLORFUL_COLORS)
        //barDataSet.color =
        resources.getColor(R.color.browser_actions_bg_grey)

        barChart.legend.calculatedLabelSizes

        barChart.animateY(2000)

        buttonGrafica.text = "Grafica Participacion"
    }
}

buttonInsertar.setOnClickListener {

    var media = 0.0f
    var tamaño = 0.0f

    var i = 0
    Eliminar(idAsignatura)
    Guardar("DNI;Nombre;", idAsignatura)
    for (j in aTitulo) {
        Guardar("$j;", idAsignatura)
    }
    Guardar("Nota Media\n", idAsignatura)
    while (i < adni.size) {
        var dni = adni[i]
        Guardar(dni + ";" + anombre[i] + ",", idAsignatura)
        for (id in aCuestionarios) {
            control = false
            for (cobjeto in notas) {
                if (id == cobjeto.id && dni == cobjeto.dni) {
                    Guardar(cobjeto.nota.toString() + ";",
                        idAsignatura)

                    media = cobjeto.nota
                    tamaño++
                    control = true
                }
            }
        }
        i++
    }
}

```

```

    }
    }
    if (control == false) {
        Guardar(";", idAsignatura)
    }
}

media = media / aCuestionarios.size
Guardar(media.toString() + "\n", idAsignatura)

i++
}

//Todo:
*****
try {
    val emails = arrayOf(email)
    val rutaSd =
baseContext.getExternalFilesDir(null)?.absolutePath
    val miCarpeta = File(rutaSd, "Notas")
    val ficheroFisico = File(miCarpeta,
"$idAsignatura.csv")
    val uri = FileProvider.getUriForFile(this
,
this.getApplicationContext().getPackageName() +
".provider",
    ficheroFisico
)
    val intent = Intent(Intent.ACTION_SEND).apply {
        type = "*/*"
        putExtra(Intent.EXTRA_EMAIL, emails)
        putExtra(
            Intent.EXTRA_TEXT,
            "En este correo se encuentran todas las notas
de la asignatura $nombre.\n\nEste correo ha sido " +
            "enviado desde la aplicacion QUIZU"
        )
        putExtra(
            Intent.EXTRA_SUBJECT,
            "QUIZU: Exportación de notas de $nombre"
        )
    }
    intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION)

    putExtra(Intent.EXTRA_STREAM, uri)
}

startActivity(intent)
}catch (e: Exception){
    Toast.makeText(

```

```
        this,
        "No se ha podido enviar el archivo",
        Toast.LENGTH_LONG
    ).show()
    showAlert(e.toString())
}

}

}

}

}

}

fun showAlert(mensaje: String){
    val builder = AlertDialog.Builder(this)
    builder.setTitle("Error")
    builder.setMessage(mensaje)
    builder.setPositiveButton("Aceptar", null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

private fun controlNotas(idAsignatura: String) {

}

}
```

3 Ficheros de interfaz

3.1 activity_asignatura_alumno.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"

    tools:context=".AsignaturaAlumno">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/scrollView2"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <ImageView
            android:id="@+id/imageView5"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_marginLeft="16dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginRight="16dp"
            android:layout_marginBottom="16dp"
            android:src="@mipmap/logo"

            app:layout_constraintBottom_toTopOf="@+id/textViewNombreAsignaturaAlumno"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/textViewNombreAsignaturaAlumno"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
            android:layout_marginLeft="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginRight="16dp"
            android:layout_marginBottom="20dp"
            android:text="Nombre asignatura"
            android:textSize="22dp"
            app:layout_constraintBottom_toTopOf="@+id/linearLayout"
            app:layout_constraintEnd_toEndOf="parent"
```

```

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView5" />

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="10dp"
    android:orientation="horizontal"

    app:layout_constraintBottom_toTopOf="@+id/buttonCuestionariosAlumno"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintTop_toBottomOf="@+id/textViewNombreAsignaturaAlumno">

    <Button
        android:id="@+id/buttonRealizarCuestionarioAlumno"
        android:layout_width="157dp"
        android:layout_height="match_parent"
        android:backgroundTint="#FF8400"
        android:text="Realizar Cuestionario" />

    <EditText
        android:id="@+id/editTextCodigoCuestionarioAlumno"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_marginLeft="20dp"
        android:layout_weight="1"
        android:ems="10"
        android:hint="Introduce código"
        android:inputType="number"
        android:paddingTop="12dp" />

</LinearLayout>

<Button
    android:id="@+id/buttonCuestionariosAlumno"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:backgroundTint="#FF8400"
    android:text="Cuestionarios"
    app:layout_constraintBottom_toTopOf="@+id/space3"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />

<Space
    android:id="@+id/space3"
    android:layout_width="0dp"

```

```

        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="10dp"

app:layout_constraintBottom_toTopOf="@+id/buttonHomeAsignaturaAlumno"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/buttonCuestionariosAlumno" />

<Button
    android:id="@+id/buttonHomeAsignaturaAlumno"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="61dp"
    android:backgroundTint="#FF8400"
    android:text="Home"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/space3" />

</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

3.2 activity_asignatura_profesor.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".AsignaturaProfesor">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/scrollView6"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

```



```
<LinearLayout
    android:id="@+id/linearLayout4"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/logo" />

    <TextView
        android:id="@+id/textViewNombreAsignaturaProfesor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Nombre asignatura"
        android:textSize="22dp" />

    <TextView
        android:id="@+id/textViewIdAsignaturaProfesor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="idAsignatura" />

    <Button
        android:id="@+id/buttonCuestionariosLanzados"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="#FF8400"
        android:text="Cuestionarios Activos" />

    <Button
        android:id="@+id/buttonCuestionariosProfesor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="#FF8400"
        android:text="Cuestionarios" />

    <Button
        android:id="@+id/buttonNotasExportar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="#FF8400"
        android:text="Exportar Notas" />

    <Button
        android:id="@+id/buttonListaAlumnosAsignatura"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_marginTop="10dp"
        android:backgroundTint="#FF8400"
        android:text="Alumnos" />

<Space
    android:layout_width="match_parent"
    android:layout_height="18dp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="35dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonCrearCuestionarioAsignatura"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Crear Cuestionario" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonCrearPregutna"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Crear Pregunta" />

    <Space
        android:layout_width="20dp"
        android:layout_height="match_parent" />

    <Button
        android:id="@+id/buttonPreguntas"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Banco de preguntas" />
</LinearLayout>

<Space
    android:layout_width="match_parent"
    android:layout_height="35dp" />

<Button
    android:id="@+id/buttonHomeAsignaturaProfesor"
    android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:backgroundTint="#FF8400"
        android:text="Home" />

    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

3.3 activity_auth.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/authLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".AuthActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="10dp"
            android:layout_marginEnd="10dp"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical">

                <ImageView
                    android:id="@+id/imageView"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="44dp"
                    app:srcCompat="@mipmap/logo" />

                <EditText
                    android:id="@+id/editTextEmail"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="48dp"
                    android:ems="10"
                    android:hint="Usuario"
                    android:inputType="textEmailAddress" />

                <EditText
                    android:id="@+id/editTextPassword"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:hint="Contraseña"
        android:inputType="textPassword" />

<Button
    android:id="@+id/accederButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:backgroundTint="#FF8400"
    android:text="Acceder"
    android:textColorLink="#FFFFFF" />

<Button
    android:id="@+id/registrarResult"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:backgroundTint="#FF8400"
    android:text="Registrarse" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:orientation="horizontal">

    <Space
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1" />

    <Space
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1" />

</LinearLayout>

</LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.4 activity_editar_perfil.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/bg_rn"
tools:context=".editarPerfil">

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/scrollView5"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout

        android:id="@+id/linearlayout3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:srcCompat="@mipmap/logo" />

        <TextView
            android:id="@+id/bienvenidaModify"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="Rellene los valores que desee modificar:"
            android:textSize="24dp" />

        <EditText
            android:id="@+id/editTextName2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:ems="10"
            android:hint="Nombre Completo"
            android:inputType="textPersonName" />

        <Space
            android:layout_width="match_parent"
            android:layout_height="399dp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal">
```

```

        <Button
            android:id="@+id/buttonCancelarModificarPerfil"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Cancelar" />

        <Space
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1" />

        <Button
            android:id="@+id/buttonAceptarModificarPerfil"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Aceptar" />
    </LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.5 activity_form_asignatura.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".FormAsignatura">

    <LinearLayout
        android:id="@+id/constraintLayout"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <ImageView
            android:id="@+id/imageView3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

```

```
        android:layout_marginBottom="100px"
        app:srcCompat="@mipmap/logo" />

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Rellene lo siguiente campos" />

<EditText
    android:id="@+id/editTextNombreAsignatura"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nombre de la asignatura"
    android:inputType="textPersonName" />

<LinearLayout
    android:id="@+id/linearLayout5"
    android:layout_width="match_parent"
    android:layout_height="61dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="259dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Introduzca curso académico:" />

    <EditText
        android:id="@+id/editTextAnoAcademico"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:hint="aa/aa"
        android:inputType="textPersonName"
        android:textAlignment="center" />
</LinearLayout>

<LinearLayout
    android:id="@+id/linearLayout6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonCancelarFormAsignatura"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Cancelar" />

    <Space
        android:layout_width="wrap_content"
```

```

        android:layout_height="match_parent"
        android:layout_weight="1" />

        <Button
            android:id="@+id/buttonCrearFormAsignatura"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Crear" />
    </LinearLayout>

</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

3.6 activity_form_question.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".FormQuestion">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginStart="16dp"
                android:layout_marginTop="16dp"
                android:layout_marginEnd="16dp"
                android:layout_marginBottom="16dp"
                android:orientation="vertical">

                <ImageView
                    android:id="@+id/imageView6"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="100px"
                    app:srcCompat="@mipmap/logo" />

                <TextView
                    android:id="@+id/textView6"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"

```



```
        android:text="Rellene los siguientes campos: " />

<EditText
    android:id="@+id/editTextEnunciadoForm"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:ems="10"
    android:hint="Introduzca enunciado de la pregunta"
    android:inputType="textPersonName" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="Introduzca las posibles respuestas." />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <CheckBox
        android:id="@+id/checkboxOpcion1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="@null"

        />

    <EditText
        android:id="@+id/editTextFromOpcion1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:ems="10"
        android:hint="Opcion 1"
        android:inputType="textPersonName" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <CheckBox
        android:id="@+id/checkboxOpcion2"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="@null" />

    <EditText
        android:id="@+id/editTextFromOpcion2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:ems="10"
        android:hint="Opcion 2"
        android:inputType="textPersonName" />
```

```
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <CheckBox
        android:id="@+id/checkBoxOpcion3"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="@null"
    />

    <EditText
        android:id="@+id/editTextFromOpcion3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:ems="10"
        android:hint="Opcion 3"
        android:inputType="textPersonName" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <CheckBox
        android:id="@+id/checkBoxOpcion4"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="@null"/>

    <EditText
        android:id="@+id/editTextFromOpcion4"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:ems="10"
        android:hint="Opcion 4"
        android:inputType="textPersonName" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Numero de la respuesta correcta" />

    <EditText
        android:id="@+id/editTextCorrecta"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Valor entre 1 y 4"
        android:inputType="textPersonName"
        android:textAlignment="center" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/buttonAnteriorFormQuestion"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Cancelar" />

        <Space
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1" />

        <Button
            android:id="@+id/buttonAceptarFormQuestion"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Aceptar" />
    </LinearLayout>

</LinearLayout>
</LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.7 activity_form_quiz.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".FormQuiz">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/scrollView4"
        android:layout_width="0dp"

```

```

    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="12dp"
        android:layout_marginEnd="12dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <ImageView
                android:id="@+id/imageView7"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="15px"
                app:srcCompat="@mipmap/logo" />

            <TextView
                android:id="@+id/introFormQuiz"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Introduzca los siguientes datos para
comenzar la creacion del cuestionario:" />

            <EditText
                android:id="@+id/editTextTituloQuiz"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="16dp"
                android:ems="10"
                android:hint="Introduzca título del cuestionario"
                android:inputType="textPersonName" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/textView2"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"

                    android:text="Introduzca penalizacion al fallo:" />

```

```

        <EditText
            android:id="@+id/editTextPenalty"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="Entre 0 y 100 (%)"
            android:inputType="number"
            android:textAlignment="center" />

    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonAñadirPreguntaFormQuiz"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Añadir Pregunta" />

    <Space
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <Button
        android:id="@+id/buttonCrearPreguntaFormQuiz"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Crear Pregunta" />
</LinearLayout>

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10px"
    android:layout_marginBottom="10px"
    android:text="Lista de Preguntas"
    android:textSize="16dp" />

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ListView
            android:id="@+id/listPreguntasQuiz"
            android:layout_width="match_parent"
            android:layout_height="240dp" />
    </LinearLayout>
</ScrollView>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonCancelarFormQuiz"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Cancelar" />

    <Space
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <Button
        android:id="@+id/buttonCrearFormQuiz"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FF8400"
        android:text="Crear" />
    </LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.8 activity_home.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".HomeActivity">

    <androidx.constraintlayout.widget.ConstraintLayout

```

```

    android:id="@+id/scrollView3"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout
        android:id="@+id/linearLayout7"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginStart="16dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginBottom="16dp"
            android:orientation="vertical">

            <ImageView
                android:id="@+id/imageView4"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                app:srcCompat="@mipmap/logo" />

            <TextView
                android:id="@+id/nametextView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="16dp"
                android:text="Nombre"
                android:textSize="24dp" />

            <Button
                android:id="@+id/buttonCrearAsignatura"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:backgroundTint="#FF8400"
                android:text="Crear Asignatura" />

            <Button
                android:id="@+id/buttonAsignaturas"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:backgroundTint="#FF8400"
                android:text="Asignaturas" />

            <Space
                android:layout_width="match_parent"
                android:layout_height="243dp" />

            <Button

```

```

        android:id="@+id/buttonEditarPerfil"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:backgroundTint="#FF8400"
        android:text="Editar perfil" />

<Button
    android:id="@+id/cerrarSesion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:backgroundTint="#FF8400"
    android:text="Cerrar Sesion" />

<Button
    android:id="@+id/buttonSalirHome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:backgroundTint="#FF8400"
    android:text="Salir" />

</LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.9 activity_home_alumno.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".HomeAlumno">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/scrollView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:id="@+id/linearLayout8"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            app:layout_constraintBottom_toBottomOf="parent"

```



```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@mipmap/logo" />

    <TextView
        android:id="@+id/nametextViewAlumno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Nombre"
        android:textSize="24dp" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/buttonRealizarCuestionarioHomeAlumno"
            android:layout_width="157dp"
            android:layout_height="match_parent"
            android:backgroundTint="#FF8400"
            android:text="Realizar Cuestionario" />

        <EditText
            android:id="@+id/editTextCodigoCuestionarioAlumno"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_marginLeft="20dp"
            android:layout_weight="1"
            android:ems="10"
            android:hint="Introduce código"
            android:inputType="number"
            android:paddingTop="12dp" />

    </LinearLayout>

    <Button
        android:id="@+id/buttonMisAsignaturas"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="#FF8400"
        android:text="Mis Asignaturas" />

<Button
    android:id="@+id/buttonAsignaturasAlumno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:backgroundTint="#FF8400"
    android:text="Asignaturas" />

<Space
    android:layout_width="match_parent"
    android:layout_height="205dp" />

<Button
    android:id="@+id/buttonEditarPerfilAlumno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:backgroundTint="#FF8400"
    android:text="Editar perfil" />

<Button
    android:id="@+id/cerrarSesionHomeAlumno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:backgroundTint="#FF8400"
    android:text="Cerrar Sesion" />

</LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.10 activity_lista.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".Lista">

    <ScrollView
        android:id="@+id/scrollView8"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"

```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView8"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="100px"
        app:srcCompat="@mipmap/logo" />

    <TextView
        android:id="@+id/textViewLista"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView" />

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="189dp"
        android:layout_marginTop="20dp">

    </ListView>

    <ScrollView
        android:id="@+id/marg"
        android:layout_width="match_parent"
        android:layout_height="201dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/textViewNotaMedia"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/textViewDesplegadoLista"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:textSize="20sp" />

        </LinearLayout>
    </ScrollView>
```

```

        <Button
            android:id="@+id/buttonNotas"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:backgroundTint="#FF8400"
            android:text="Notas" />

        <Button
            android:id="@+id/buttonSeleccionar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:backgroundTint="#FF8400"
            android:text="Seleccionar" />

    </LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.11 activity_notas.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".Notas">

    <ScrollView
        android:id="@+id/scrollView9"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <com.github.mikephil.charting.charts.BarChart
                android:id="@+id/barChart"
                android:layout_width="match_parent"
                android:layout_height="582dp" />

            <Button

```

```

        android:id="@+id/buttonInsertar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF8400"
        android:text="Exportar" />

        <Button
            android:id="@+id/buttonGrafica"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="15px"
            android:backgroundTint="#FF8400"
            android:text="Grafica Participacion" />

    </LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

3.12 activity_registro.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".Registro">

    <ScrollView
        android:id="@+id/scrollView10"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginStart="16dp"
                android:layout_marginTop="16dp"
                android:layout_marginEnd="16dp"
                android:layout_marginBottom="16dp"
                android:orientation="vertical">

                <ImageView
                    android:id="@+id/imageView2"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        app:srcCompat="@mipmap/ic_launcher" />

<TextView
    android:id="@+id/bienvenida"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Rellene los siguientes datos para completar
el registro:"

    android:textSize="24dp" />

<EditText
    android:id="@+id/editTextSignEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:ems="10"
    android:hint="Email"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/editTextName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:hint="Nombre Completo"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/editTextDNI"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="DNI"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/editTextSignPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:hint="Contraseña"
    android:inputType="textPassword" />

<EditText
    android:id="@+id/editTextConfirmSignPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:hint="Confirme contraseña"
    android:inputType="textPassword" />

```

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RadioButton
        android:id="@+id/radioButtonProfesor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Profesor"
        android:textSize="20dp" />

    <RadioButton
        android:id="@+id/radioButtonAlumno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Alumno"
        android:textSize="20dp" />

</RadioGroup>

<Space
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:gravity="top" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonSignCancelar"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF8400"
        android:text="Cancelar" />

    <Space
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1" />

    <Button
        android:id="@+id/buttonSignAceptar"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF8400"
        android:text="Registrarse" />

</LinearLayout>

</LinearLayout>
</LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

3.13 activity_seleccionado.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".Seleccionado">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"

        android:orientation="vertical"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="16dp">

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical" >

                <ImageView
                    android:id="@+id/imageView4"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    app:srcCompat="@mipmap/logo" />

                <TextView
                    android:id="@+id/textViewSeleccionadoArriba"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="20dp"
                    android:textSize="21sp" />

                <ScrollView
                    android:layout_width="match_parent"
                    android:layout_height="240dp">

                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:orientation="vertical">
```



```

        <ListView
            android:id="@+id/listViewSeleccionado"
            android:layout_width="match_parent"
            android:layout_height="210dp"
            android:layout_marginTop="30dp">

        </ListView>

    </LinearLayout>
</ScrollView>

<TextView
    android:id="@+id/textViewSelccionadoAbajo"
    android:layout_width="match_parent"
    android:layout_height="139dp"
    android:layout_weight="1"
    android:textSize="21sp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:layout_weight="1"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonAtrasSeleccionado"
        android:layout_width="163dp"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF8400"
        android:text="Volver" />

    <Space
        android:layout_width="38dp"
        android:layout_height="match_parent" />

    <Button
        android:id="@+id/buttonEliminarSeleccionado"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF8400"
        android:text="Eliminar" />

</LinearLayout>

<Button
    android:id="@+id/buttonLanzarCuestionario"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:backgroundTint="#FF8400"
    android:text="Activar Cuestionario" />

</LinearLayout>
</ScrollView>

</LinearLayout>

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

3.14 activity_solve_quiz.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bgrn"
    tools:context=".SolveQuiz">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" >

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginStart="16dp"
                android:layout_marginTop="16dp"
                android:layout_marginEnd="16dp"
                android:layout_marginBottom="16dp"
                android:orientation="vertical">

                <ImageView
                    android:id="@+id/imageView9"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginBottom="100px"
                    app:srcCompat="@mipmap/logo" />

                <TextView
                    android:id="@+id/textViewTituloQuiz"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="Titulo del cuestionario"
                    android:textSize="20dp" />

                <TextView
                    android:id="@+id/textViewNumeroPregunta"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="28dp"
                    android:text="Pregunta  x" />

                <TextView
                    android:id="@+id/textViewEnunciadoSolve"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Enunciado de la pregunta"

        android:textColor="@color/design_default_color_on_secondary"
        android:textSize="18dp" />

    <RadioGroup
        android:id="@+id/radioGroupQuiz"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp">

        <RadioButton
            android:id="@+id/radioButtonOpcion1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Opcion 1"

            android:textColor="?android:attr/textColorPrimaryDisableOnly" />

        <RadioButton
            android:id="@+id/radioButtonOpcion2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Opcion 2" />

        <RadioButton
            android:id="@+id/radioButtonOpcion3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Opcion 3" />

        <RadioButton
            android:id="@+id/radioButtonOpcion4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Opcion 4" />
    </RadioGroup>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="35dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/buttonPreguntaAnterior"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Anterior" />

        <Space
            android:layout_width="0dp"
            android:layout_height="wrap_content"
```

```
        android:layout_weight="1" />

        <Button
            android:id="@+id/buttonAvanzar"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Siguiente" />
    </LinearLayout>

    <Space
        android:layout_width="match_parent"
        android:layout_height="39dp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <Space
            android:layout_width="250dp"
            android:layout_height="match_parent"
            android:layout_weight="1" />

        <Button
            android:id="@+id/buttonCancelar"
            android:layout_width="125dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:backgroundTint="#FF8400"
            android:text="Cancelar" />
    </LinearLayout>

</LinearLayout>
</LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

4 Ficheros de configuración

4.1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.quizu">

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:requestLegacyExternalStorage="true"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Quizu"
        android:usesCleartextTraffic="true">
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="com.example.quizu.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths" />
            </provider>

            <activity android:name=".Notas"
                android:screenOrientation="portrait"></activity>
            <activity android:name=".HomeAlumno"
                android:screenOrientation="portrait"/>
            <activity android:name=".AsignaturaAlumno"
android:screenOrientation="portrait"/>
            <activity android:name=".FormAsignatura"
android:screenOrientation="portrait"/>
            <activity android:name=".Seleccionado"
android:screenOrientation="portrait"/>
            <activity android:name=".editarPerfil"
android:screenOrientation="portrait"/>
            <activity android:name=".AsignaturaProfesor"
android:screenOrientation="portrait"/>
            <activity android:name=".Lista"
```

```

    android:screenOrientation="portrait"/>
    <activity
        android:name=".FormQuestion"
        android:windowSoftInputMode="adjustResize"
        android:screenOrientation="portrait"/>
    <activity
        android:name=".SolveQuiz"
        android:windowSoftInputMode="adjustResize"
        android:screenOrientation="portrait"/>
    <activity
        android:name=".FormQuiz"
        android:windowSoftInputMode="adjustResize"
        android:screenOrientation="portrait"/>
    <activity
        android:name=".Registro"
        android:windowSoftInputMode="adjustResize"
        android:screenOrientation="portrait"/>
    <activity
        android:name=".HomeActivity"
        android:windowSoftInputMode="adjustResize"
        android:screenOrientation="portrait"/>
    <activity
        android:name=".AuthActivity"
        android:theme="@style/SplashTheme"
        android:windowSoftInputMode="adjustResize"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>

```

4.2 build.gradle (Project)

```

// Top-level build file where you can add configuration options common to all
// sub-projects/modules.
buildscript {
    ext.kotlin_version = "1.4.32"
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:4.1.3"
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.3.5'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

```

```

allprojects {
    repositories {
        google()
        jcenter()
        maven { url 'https://jitpack.io' }
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}

```

4.3 build.gradle (App)

```

apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'com.google.gms.google-services'
apply plugin: 'kotlin-android-extensions'

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.quizu"
        minSdkVersion 23
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
}

dependencies {
    implementation 'com.github.PhilJay:MPAndroidChart:v2.2.4'

    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.3.2'
}

```

```
implementation 'androidx.appcompat:appcompat:1.2.0'
implementation 'com.google.android.material:material:1.3.0'
implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
implementation 'androidx.navigation:navigation-fragment:2.3.4'
implementation 'androidx.navigation:navigation-ui:2.3.4'
implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.3.1'
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.3.1'
implementation 'androidx.navigation:navigation-fragment-ktx:2.3.4'
implementation 'androidx.navigation:navigation-ui-ktx:2.3.4'
implementation 'com.google.firebase:firebase-firestore-ktx:22.1.2'
testImplementation 'junit:junit:4.+'
implementation 'com.google.code.gson:gson:2.8.6'
implementation 'com.android.volley:volley:1.1.1'
implementation platform('com.google.firebase:firebase-bom:27.1.0')
implementation 'com.google.firebase:firebase-database-ktx'
implementation 'com.google.firebase:firebase-firestore-ktx'
androidTestImplementation 'androidx.test.ext:junit:1.1.2'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
implementation platform('com.google.firebase:firebase-bom:27.1.0')
implementation 'com.google.firebase:firebase-analytics-ktx'
implementation platform('com.google.firebase:firebase-bom:26.8.0')
implementation 'com.google.firebase:firebase-auth'
implementation platform('com.google.firebase:firebase-bom:26.8.0')
implementation 'com.google.firebase:firebase-auth-ktx'
implementation 'com.google.android.gms:play-services-auth:19.0.0'
}
```


5 Otros ficheros

5.1 Ficheros de objetos

5.1.1 Cuestionario.kt

```
package com.example.quizu

data class Cuestionario (

    val nota: Float,
    val id: String,
    val dni: String,
    val titulo: String,
)
```

5.1.2 Pregunta.kt

```
package com.example.quizu

data class Pregunta (

    val idPregunta: String,
    val enunciado: String,
    val numeroRespuestas: String,
    val opcion1: String,
    val opcion2: String,
    val opcion3: String,
    val opcion4: String,
    val correcta: Int
)
```

5.2 Ficheros de interfaz complementarios

5.2.1 Splash.xml

Fichero usado para la creación de la animación al iniciar la aplicación.

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@color/teal_200"/>
    <item>
        <bitmap android:src="@mipmap/logo" android:gravity="center"/>
    </item>
</layer-list>
```

5.2.2 Bgrn.xml

Fichero usado para la creación del fondo de todas las interfaces.

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:startColor="#60FFEB3B"
        android:endColor="#A1FF9800"
        android:angle="270"

    ></gradient>

</shape>
```

5.2.3 Colors.xml

Fichero modificado para usar los colores de la aplicación.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#60FFEB3B</color>
    <color name="purple_500">#60FFEB3B</color>
    <color name="purple_700">#60FFEB3B</color>
    <color name="teal_200">#FF9800</color>
    <color name="teal_700">#FF9800</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
</resources>
```