

17/02/2021

hoy vamos a hablar de UX (experiencia de usuario) y UI (interfaz de usuario).

para saber si un dominio esta reservado nos metemos en IONOS y para localizar marcas y registrarlas hay que mirar en el OEPN. En España no se puede patentar software por lo que tendremos que patentarlo como otra cosa, y lo aconsejable es hacerlo como si fuera un libro en ISBN.

Para subir una app tanto a Google Play como a Apple Store tenemos que tener un mail de soporte (para que nos escriban cuando haya algún problema) y también necesitamos la política de privacidad.

18/02/2021

Hoy empezamos con programación, pero solo teoría.

Yo voy a programar para hacer programas dentro de un dispositivo, que tiene 2 cosas fundamentales: una memoria RAM y una CPU. Cuando programamos no dependemos del hardware (podemos hacer un programa para un móvil, un ordenador, un coche...).

Un algoritmo son los pasos que hace un programa para cumplir su función.

3 tipos de lenguajes: bajo nivel (10110000 binario, lenguaje ensamblador y A2), medio nivel (C) y lenguajes de alto nivel (Java, C++, Cobol, PHP, Python...). Los lenguajes de bajo nivel (lenguaje máquina) que son muy problemáticos porque solo se aplican a una máquina concreta, no a cualquiera, es decir si hacemos una app en lenguaje máquina para un iPhone 12 solo se ejecuta única y exclusivamente en los iPhone 12, no en los iPhone 11 ni en los iPhone 6.

VMWARE es una máquina virtual que te permite replicar distintos sistemas operativos. Usando VMware puedo programar una app para iPhone en un Windows, pero no es lo óptimo porque me va a comer una cantidad salvaje de RAM y me va a ir fatal el ordenador.

Mirar qué es web app, app nativa, app no nativa y app híbrida.

MIRAR LENGUAJE XAMARIN.

Mirar qué son los paradigmas de programación.

DIAGRAMA DE FLUJO PARA SABER SI UN NÚMERO ES MENOR, IGUAL O MAYOR QUE 5.

A la hora de hacer un programa con condiciones tenemos que procurar hacer las preguntas de la forma más óptima para que el programa sea lo más eficiente posible.

```
Let arr = [1,2,3,4,5,6,7,8,9,10,11,12]
```

```
For (i=0;i<10;i++){  
    let mul2=arr[i]*2  
    console.log(mul2)  
}
```

19/02/2021

Para que la CPU ejecute el código, el código tiene que ser traducido para que la máquina lo entienda. Para ello, al ejecutar el código, este tiene que pasar por un intérprete (traductor), un ensamblador o una máquina virtual.

Aplicación nativa

La aplicación nativa está desarrollada y **optimizada específicamente para el sistema operativo** determinado y la plataforma de desarrollo del fabricante (Android, iOS, etc).

Este tipo de aplicaciones **se adapta al 100% con las funcionalidades y características del dispositivo** obteniendo así una **mejor experiencia de uso**. Sin embargo, el desarrollo de una aplicación nativo comporta un **mayor coste**, puesto que si se desea realizar una aplicación multiplataforma se ha de realizar una nueva versión para cada sistema operativo, multiplicando así los costes de desarrollo.

Algunos **ejemplos** de aplicación nativa, serían **Whatsapp o Facebook**.

Aplicación web

La aplicación web es la opción **más sencilla y económica** de crear aplicaciones, puesto que al desarrollar una única aplicación se reducen al máximo los costes de desarrollo. Asimismo, en este tipo de aplicaciones, puede utilizarse el “**responsive web design**”, creando así una única aplicación adaptada para todo tipo de dispositivos. Por el contrario, la aplicación web ofrece una **peor experiencia de uso**, puesto que ignora las características del dispositivo y una **menor seguridad** ya que depende de la seguridad que ofrezca el propio navegador.

Aplicación híbrida

Este tipo de aplicación **aprovecha al máximo la versatilidad de un desarrollo web y tiene la capacidad de adaptación al dispositivo como una app nativa**. Permite utilizar los estándares de desarrollo web (HTML5) y aprovechar las funcionalidades del dispositivo tales como la cámara, el GPS o los contactos. Además, comporta un **menor coste** que una aplicación nativa y una **mejor experiencia de uso** que una aplicación web. Sin embargo, tiene un **rendimiento ligeramente inferior** al de una aplicación nativa debido a que cada página debe ser renderizada desde el servidor y supone una **mayor dificultad de desarrollo**.

link - <https://www.raona.com/aplicacion-nativa-web-hibrida/>

En las pantallas de los móviles se ve en una escala de colores que se llaman colores RGB.

22/02/2021

Todas las pantallas que vemos en Android se llama “activity”. Crear las apps en Android 5.0 lollipop (es la version mas estable y funciona en la mayoria de dispositivos sin problemas).

XML -> es como una mezcla de HTML y CSS.

SITIO DONDE ESTAN LAS ETIQUETAS DE CSS -> <https://www.w3schools.com/css/>

Las medidas de letra en css NO se ponen en px, SI se ponen en dp o sp para que se adapte al dispositivo.

23/02/2021

Cuando queremos cambiar los textos a otro idioma hay que crear primero un directorio AL MISMO NIVEL que el otro archivo de string. O sea, que los strings de todos los idiomas tienen que estar al mismo nivel.

Para crear un string de otro idioma hay que darle al repositorio “res” y ahí, click derecho y darle a android resource directory y ponerle el valor “string-XX” siendo XX el sufijo el correspondiente al país (los sufijos de los países se buscan en google en ISO CODE COUNTRY).

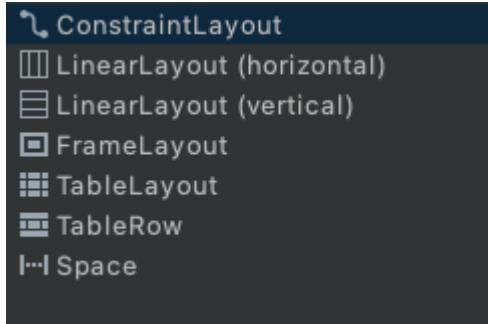
Luego, crear el string dándole click derecho encima de la carpeta de strings y a create new resource file y en directory name poner el nombre del directorio correspondiente a su idioma.

COSAS IMPORTANTES PARA LOS CONSTRAINTS

las preguntas van a ir relacionadas con los layout.

¿Qué tipos de layout puede salir? referal layout NO ES NUNCA, YA NO EXISTE.

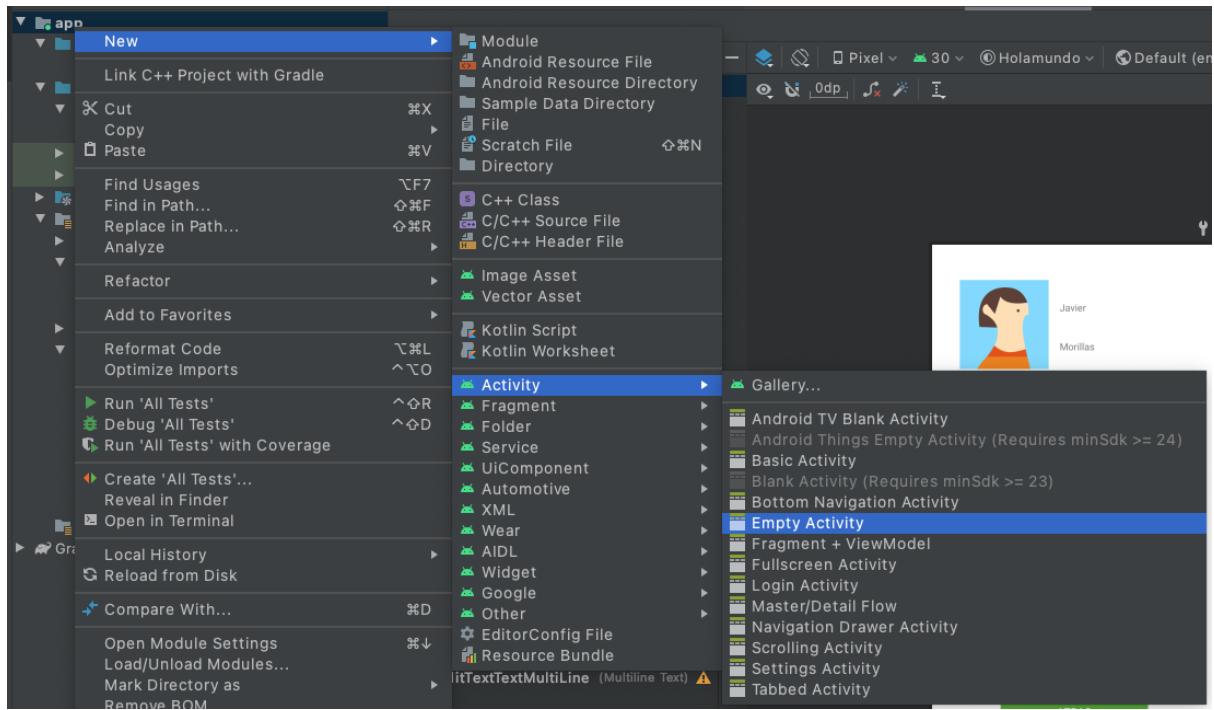
Nosotros vamos a jugar casi siempre con:



El importante es constraint layout. Constraint es bien, constraint es salud. Constraint lo que hace es posicionar los elementos de forma porcentual, como si fueran unos muelles. Tu lo que haces es colocar los elementos como quieras usando los muelles del constraint y esa posición relativa la coloca adaptandola al tamaño y posición del dispositivo.

24/02/2021

para crear bien un nuevo activity hay que hacerlo asi:



Y luego ponerle un nombre adecuado, de esta forma se crea la nueva activity ya vinculada para poder enlazar las cosas mas facilmente.

Siempre que quiera hacer un cambio de pantalla (darle a un boton para que pase a otro sitio), tengo que ponerle obligatoriamente el Intent (es para que la app tambien reaccione a los botones del propio movil).

Cuando quiero usar atributos y cosas asi, lo pongo con minuscula en Java. Cuando quiero usar objetos del sistema, va en mayuscula.

25/02/2021

aqui todo fueron practicas

26/02/2021

Cuando quiero hacer eventos en java, tengo que hacer varios pasos:

1º declaro las variables que voy a usar, tengo que localizar los elementos con los que quiero interactuar.

```
// declaro cosas
EditText inputEdad = findViewById(R.id.inputEdad);
Button btnCalcular = findViewById(R.id.btnCalcularEdad);
TextView resultado = findViewById(R.id.resultado);
```

2º marco el boton que va a marcar la pauta y le digo que va a ser el disparador del evento. Como quiero que el evento se ejecute al realizar un click, tengo que usar el onClick:

```
btnCalcular.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        |
    }
});
```

Lo que hago es 1º busco por ID el boton que quiero usar, y luego con el metodo lo que hago es coger el boton que he cogido en el paso 1º y luego le convierto en un escuchador, un objeto que va a ejecutar una orden cuando se de una condicion (el click, en este caso).

3º para recoger el dato lo que hacemos es coger el dato del input y lo pasamos a un string, y luego lo pasamos a un integer. Luego hacer el resto de la operacion. En este caso ademas, le he metido un if y le he puesto que salga un aviso para cuando el usuario no ha metido un valor adecuado (ese aviso se llama Toast).

```
public void onClick(View v) {
    // declaro String y guardo el texto editado por el usuario
    // he convertido en string el numero que me han dado
    String miEdad=inputEdad.getText().toString();

    // metodo de la clase integer para de string a entero
    int miEdadEntero=Integer.parseInt(miEdad);

    // aqui tengo que crear una nueva variable para guardar
    // el resultado de la multiplicacion
    int resultadoTotal=miEdadEntero*7;

    // aqui hago el contenedor del texto que me dara el resultado en string
    String sResultado = "Tu edad en años de perro son "+resultadoTotal+" años!";

    // aqui le digo al programa que me imprima el resultado por pantalla
    resultado.setText(sResultado);
}

public void onClick(View v) {
    // declaro String y guardo el texto editado por el usuario
    // he convertido en string el numero que me han dado
    String miEdad=inputEdad.getText().toString();

    // en esta condicion le estoy preguntando "si NO esta vacio..."
    if(miEdad.isEmpty()){

        // metodo de la clase integer para de string a entero
        int miEdadEntero=Integer.parseInt(miEdad);

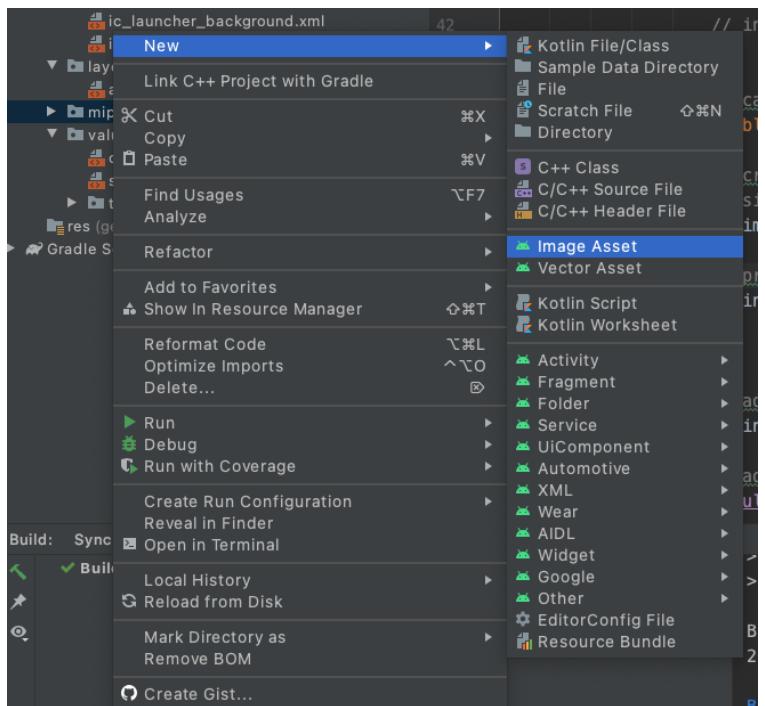
        // aqui tengo que crear una nueva variable para guardar
        // el resultado de la multiplicacion
        int resultadoTotal=miEdadEntero*7;

        // aqui hago el contenedor del texto que me dara el resultado en string
        String sResultado = "Tu edad en años de perro son "+resultadoTotal+" años!";

        // aqui le digo al programa que me imprima el resultado por pantalla
        resultado.setText(sResultado);

    } else {
        Toast.makeText(context, MainActivity.this, text:"Por favor, introduce un numero.", Toast.LENGTH_LONG).show();
    }
}
```

Para crear iconos en la app:



01/03/2021

Hay 3 tipos de métodos en java:

- protected
- private
- public

Cuando hacemos un tamaño en pixeles de las letras y las imágenes hay que tener en cuenta que:

- sp se utiliza sobre todo para textos
- dp se utiliza para todo que no son textos

Las imágenes tienen varios tipos de fondo:

- background normal
- background tint
- background tint module

Practica: Contador

```
public class MainActivity extends AppCompatActivity {

    // Dentro de la public class hago una variable
    // que va a ser donde almacene mi numero
    private int contador;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

Escribo la logica del contador:

```
// en el codigo, en parametros, hay que poner View
// porque lo que estas haciendo es usar una vista
public void aumentar(View view){
    contador++;
    mostrarContador();
};

public void reducir(View view){
    contador--;
};

public void reiniciar(View view){
    contador=0;
};

public void mostrarContador(){
    TextView contadorplus=findViewById(R.id.contador);
    contadorplus.setText(contador);
};
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    contador = 0;

    public void aumentar(){
        contador++;
    };

    public void reducir(){
        contador--;
    };

    public void reiniciar(){
        contador=0;
    };
}
```

Con este código conseguimos que el contador siempre muestre 000 mas el número que sea. Por ejemplo 099, 123, 005...

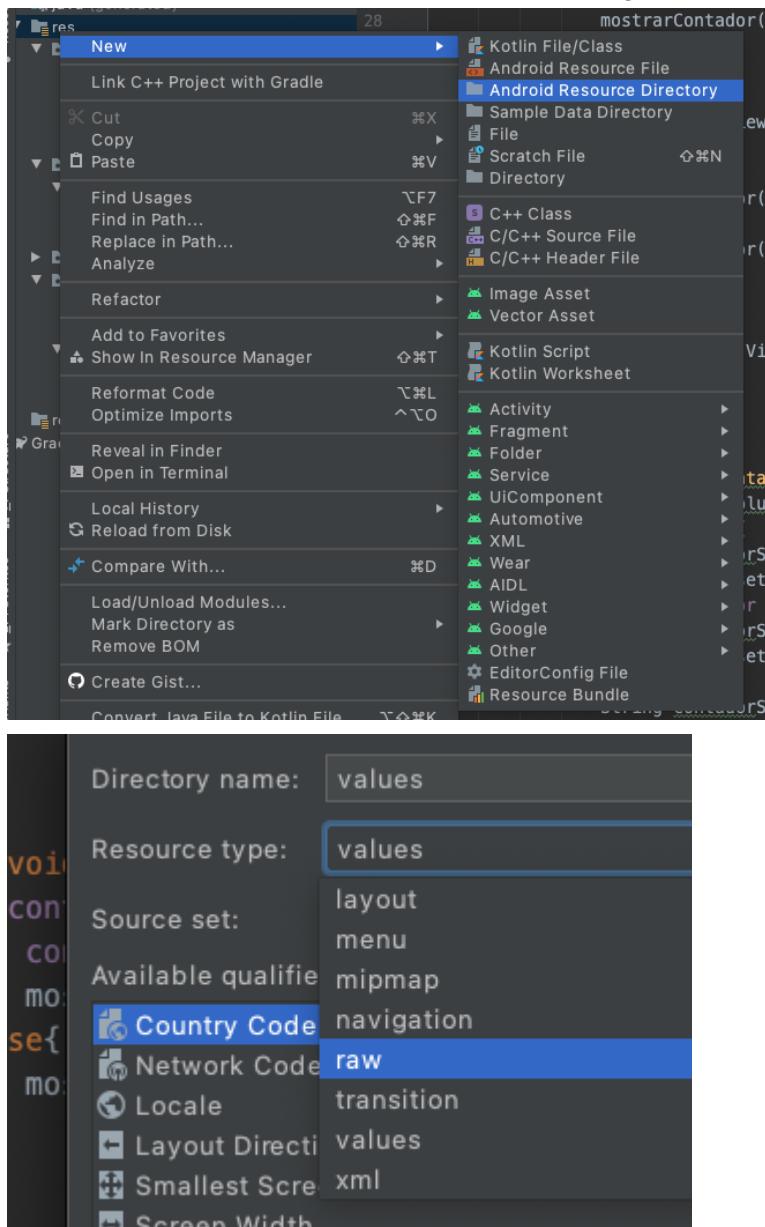
```
// Código que muestra el contador
public void aumentar(View view){
    if(contador <999){
        contador++;
        mostrarContador();
    }else {
        mostrarContador();
    };
};

public void reducir(View view){
    if(contador >0) {
        contador--;
        mostrarContador();
    }else{
        mostrarContador();
    }
};

public void reiniciar(View view){
    contador=0;
    mostrarContador();
};

public void mostrarContador(){
    TextView contadorplus=findViewById(R.id.contador);
    if(contador <10) {
        String contadorString = "00" + contador;
        contadorplus.setText(contadorString);
    } else if (contador <100){
        String contadorString = "0" + contador;
        contadorplus.setText(contadorString);
    } else{
        String contadorString = "" + contador;
        contadorplus.setText(contadorString);
    }
};
```

Siempre que quiera añadir un sonido tiene que guardarse en la carpeta RAW. Se hace así:



y para añadirle el sonido a un evento se hace así: primero creas la función que va a reproducir el sonido

```
public void playSonido( int sound){  
    MediaPlayer mp;  
    mp = MediaPlayer.create(getApplicationContext(), sound);  
    mp.start();
```

Luego le metes la funcion al evento para que reproduzca el sonido

```
public void aumentar(View view){  
    if(contador <999){  
        contador++;  
        mostrarContador();  
        playSonido(R.raw.clic);  
    }else {  
        mostrarContador();  
    };  
};  
  
public void reducir(View view){  
    if(contador >0) {  
        contador--;  
        mostrarContador();  
        playSonido(R.raw.tic);  
    }else{  
        mostrarContador();  
    };  
};
```

para copiar y pegar un texto se hace asi:

```
public void copyText (View view){  
    TextView txtcontador = findViewById(R.id.contador);  
    String valorContador = txtcontador.getText().toString();  
    ClipboardManager clipboard = (ClipboardManager) getSystemService(CLIPBOARD_SERVICE);  
    ClipData clip = ClipData.newPlainText( label: "text", valorContador);  
    clipboard.setPrimaryClip(clip);  
    Toast.makeText(getApplicationContext(), text: "Contador copiado al portapapeles", Toast.LENGTH_SHORT).show();  
};
```

02/03/2021

Hoy estamos con la practica del color RGB.

03/03/2021

Hoy estamos con la practica de la calculadora.

Vamos a hacer un evento para que escuche todos los botones y haga lo que tenga que hacer. Usaremos el setOnClickListener y los if, ya que hacer esta practica con un switch da muchos problemas; es mejor hacerlo con un if.

Usar el atributo transitionName, que es con lo que voy a usar la logica interna de la calculadora y vamos a coger el valor del transition con el metodo getTransitionName().

```
view.getText() -> con esto consigo el texto del boton  
view.getTransitionName()-> con esto consigo el valor del boton
```

La gracia de esto es que con el transition name puedo hacer botones con imagenes o con lo que sea y luego ponerle el valor de transition que quiera.

Siempre que implemento al inicio de la app un clickListener tengo que tener un metodo para que coja los botones

```
CompatActivity implements View.OnClickListener{
```

para implementar eso tengo que tener esto

```
public void onClick(View view){  
    procesarEvento(View.getId());  
}
```

tengo que hacer 4 metodos, uno para cada calculo, y asignarle un operador para la operacion y luego en el metodo resultado hacer lo que haya que hacer segundo el valor de "operador".

por ejemplo:

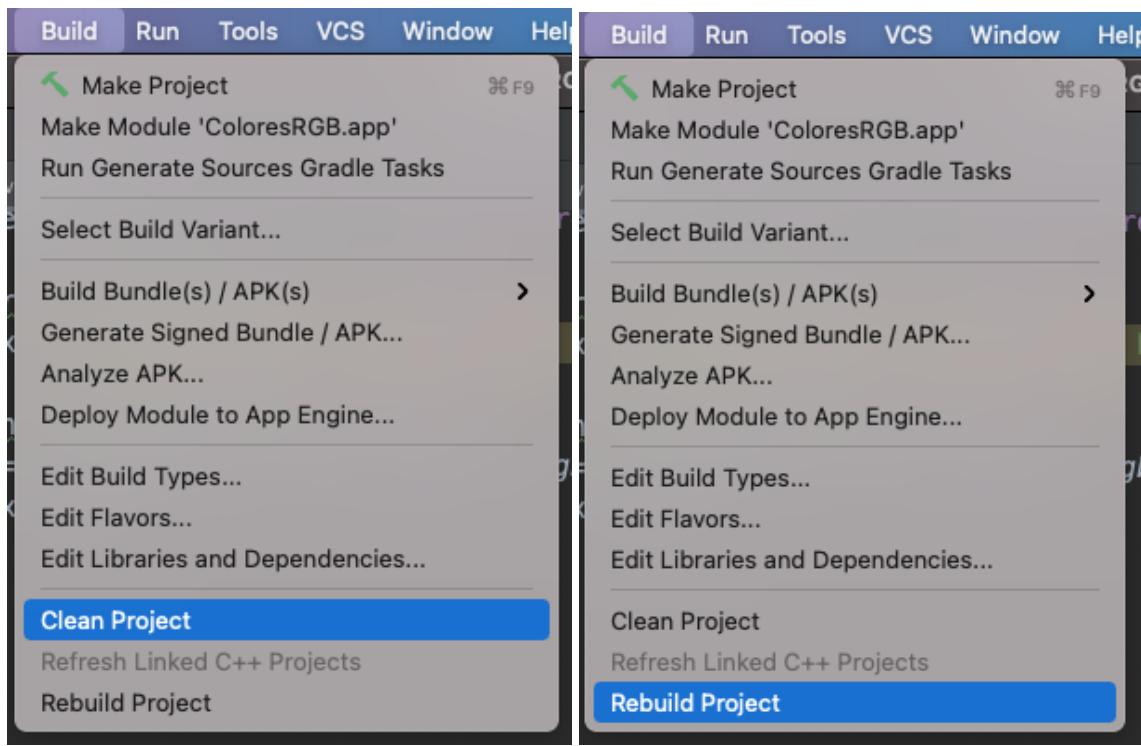
- suma ->1
- resta ->2
- multi -> 3
- div -> 4

```
metodo resultado {  
    si operador = 1  
    entonces num1 + num2  
}
```

Cuando en un metodo ponemos `(View view)` lo que queremos hacer es que detecte la vista del boton pulsado. Este metodo `public void PulsarNumero(View view){}` lo que hace es que al pulsar un numero me coge el valor de ese boton (el valor de un boton se lo he puesto con el transition name).

04/03/2021

Cuando abramos un proyecto de nuevo en casa que hayamos iniciado en clase o viceversa hay que hacer Build -> clean project y luego Build -> rebuild project.



lo suyo es tener los SDK y todo configurado con los mismos parametros en el Android Studio de todos los dispositivos que voy a usar.

hemos modificado la practica del RGB color para que cambie el color de una imagen. Para ello, usaremos que el seekbar interactue con el Tint y el TintMode.

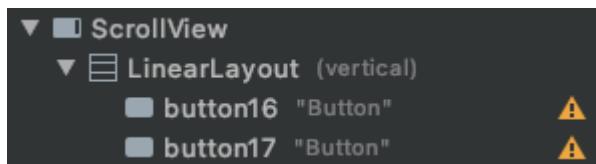
TintMode es el tipo de filtro que le quiero poner al Tint, y el Tint es el color que quiero ponerle. De todas formas, los tint y tintMode dan problemas, por lo que tenemos que usar la imagen como fondo (establecerla como fondo) y luego usar backgroundTint y backgroundTintMode.

Sobre try y catch: siempre que queramos capturar un error tenemos que hacer un LOG no un TOAST. Si el error hace que la app se cierre, pues el Toast no lo veo evidentemente, pero el log si porque me lo enseña en el logcat del android studio.

05/03/2021

Hoy miramos como usar scrollview, para cuando queremos que una pagina se muestre y podamos navegar por ella hacia arriba/abajo o izquierda/derecha.

Importante: un scrollview solo funciona con los elementos que ha sido asignado. Si yo tengo un elemento fuera del scroll, no se va a mover



Gravity es una cosa que usaremos bastante. Cuando saco elementos se me ponen automaticamente arriba a la izquierda y con gravity lo que hago es modificar la gravedad que quiero que tengan. El atributo gravity lo tienen casi todos los views y sirve para colocar las views y las cosas.

▼ gravity	▶ center_vertical
bottom	<input type="checkbox"/> false
clip_horizontal	<input type="checkbox"/> false
center	<input type="checkbox"/> false
clip_vertical	<input type="checkbox"/> false
start	<input type="checkbox"/> false
right	<input type="checkbox"/> false
center_horiz...	<input type="checkbox"/> false
fill	<input type="checkbox"/> false
fill_horizontal	<input type="checkbox"/> false
top	<input type="checkbox"/> false
left	<input type="checkbox"/> false
center_vertical	<input type="checkbox"/> false
fill_vertical	<input type="checkbox"/> false
end	<input type="checkbox"/> false

La siguiente practica va a ser un directorio sobre lo que queremos con el objetivo de luego subirlo a una base de datos.

Lo primero es crear todas las activity que queremos que tenga la app (o por lo menos las iniciales). Si se me olvida alguna, se pueden poner mas adelante. Es importante tener la estructura de la app bastante clara.

Para cambiar de una pantalla a otra tenemos que hacer un intent. Con este metodo lo que hacemos es que al darle al boton con el id btnentrar nos lleve a la activity listadoRecetas. Cuando avancemos a lo largo del proyecto lo que haremos sera usar un fragment y replicarlo para no repetir el codigo.

```
public void entrar (View view){  
    Intent intent = new Intent(getApplicationContext(), listadoRecetas.class);  
    startActivity(intent);  
}
```

Recordemos que cuando creamos una activity tenemos que hacerlo desde app -> new -> activitiv -> empty activity.

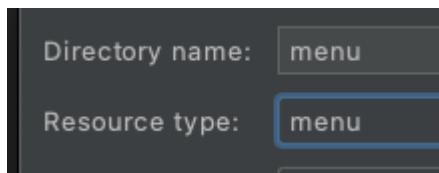
La parte de arriba de la app se llama action bar y se puede quitar (nosotros en este caso no lo haremos). Para quitarla, tenemos que poner este codigo en res->values->themes->

```
// con esto ocultamos el action bar PERO NO EL TEXTO
<item name="windowActionBar">false</item>
// con esto quitamos el titulo
<item name="windowNoTitle">true</item>
```

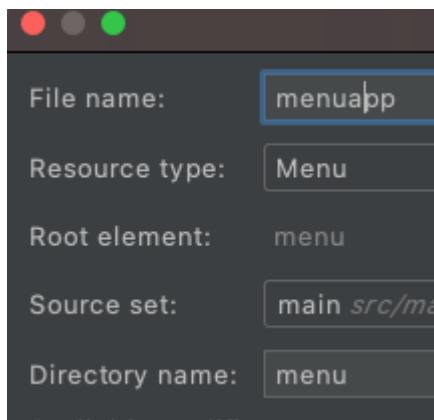
para hacer nuestro action bar personalizado se hace con la opcion toolbar en el layout. NO ES LO MISMO TOOLBAR QUE ACTION BAR. Lo que mas usaremos sera el toolbar. Al toolbar le ponemos titulo (title) y no texto (text).

Si queremos ver un codigo de menu podemos abrir un nuevo proyecto con basic activity en lugar de empty activity. La basic tiene el menu y tiene mucha basurilla, es mejor crearlo de cero.

Para crear un menu tenemos que crear un nuevo repositorio en app->new->android resource directory y le pongo eso:



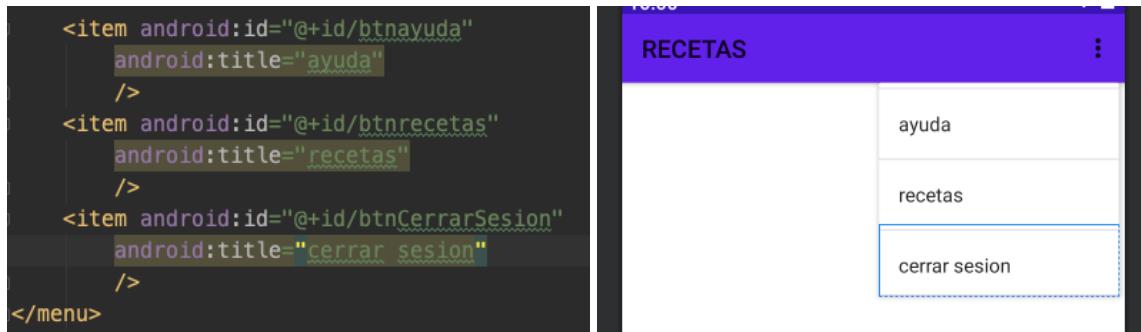
Luego hago un nuevo resource file. Se hace con res->new-> android resource file y luego aqui:



el codigo del menu tengo que escribir esto. Le pongo que el contexto es el main activitiv porque lo que quiero es que este menu de momento solo me salga en el main.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".MainActivity"
      >
</menu>
```

Para añadir botones al menu hay que hacerlo con item y cada item tiene que tener puesto minimo un id y un title.



Para crear el menu tengo que hacer este codigo:

```
// crear el menu
@Override
public boolean onCreateOptionsMenu(Menu mimenu) {
    getMenuInflater().inflate(R.menu.menuapp, mimenu);
    return true;
}
```

Esto es un clickListener para el menu, para que el menu sepa qué has tocado y te lleve a donde toque llevarte:

```
// escucha los eventos producidos en el menu
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem opcionmenu) {
    // aqui recogemos el id del boton del menu
    int id = opcionmenu.getItemId();

    if(id == R.id.btnayuda){
        //aqui estan las opcion que se ejecutan al pulsar ayuda
    } else if(id==R.id.btnrecetas){
        //opciones de recetas
    } else if (id == R.id.btnCerrarSesion){
        //opciones de cerrar sesion
    }

    return super.onOptionsItemSelected(opcionmenu);
}
```

IMPORTANTE: cuando invoco un metodo dentro de un menu no tengo que poner metodo(View view) porque el programa no tiene claro donde esta ese view. Cuando pongo un metodo(View view) tengo que ponerle (null) como parametro.

```
} else if(id==R.id.btnrecetas){  
    entrar( view: null);
```

Y este es el metodo para salir de la app y cerrarla.

```
//para finalizar la app y salir  
public void salir(View view){  
    //aqui dentro tendremos una serie de acciones  
    //para guardar distintas cosas antes de cerrar la app  
    finish();  
}
```

Cuando yo quiero subir un icono a un menu de una app puedo subirlo desde la carpeta drawable o puedo usar la categoria “android” que lo que hace es una libreria de cosas que puedo usar. En mi caso, he usado android:drawable/ para que me lleve a la libreria de imagenes y con ic lo que hago es buscar un icono. Y dentro de icono pues ya lo que me salgan.

```
    android:icon="@android:drawable/ic_menu_help"
```

Despues de asignar el icono, tengo que dar la orden de que me lo enseñe. Para ello, uso la categoria showAsAction y ahi ya pues tengo opciones: always significa que siempre va a estar ahi, ifRoom significa que siempre que haya hueco pues aparecera en la toolbar, etc...

```
    android:icon="@android:dra  
    app:showAsAction="always"  
<
```

PARA LA CALCULADORA:

Vamos a usar metodos y cosas para hacer la calculadora cientifica.

```
String frase = "Hola, como estas?";  
  
// esto me cuenta la longitud añadiendome espacios y simbolos:  
int longitud = frase.length();  
  
// para buscar una palabra dentro de una cadena:  
String punto = ".";  
int encontrarPunto = frase.indexOf(punto);  
// con esto puedo comprobar si mi frase tiene un punto o no.  
// tambien puedo usar ese metodo para decirle que empiece a contar a partir de  
// un lugar en concreto.  
int encontrarPunto2=frase.indexOf(punto, fromIndex: 5);  
  
// esto empezaria a contar empezando desde el final.  
int encontrarPunto3=frase.lastIndexOf(punto, fromIndex: 5);
```

08/03/2021

Hoy vamos a ver fragments y vamos a ver como importar y exportar cosas (vamos a hacer un react en android, basicamente).

Vamos a empezar por ponerle una toolbar a todas las activities. La actionbar se ha quedado obsoleta y descartada, hay que usar la toolbar.

Vamos a crear la toolbar de forma que solo queremos el archivo de diseño, no la activity completa. Para ello, lo creamos con click derecho sobre layout-> new resource file y le ponemos de nombre toolbar.

Para añadir la toolbar a todas las activities lo que hago es usar la etiqueta include dentro del layout que quiero importar:

```
<include layout="@layout/toolbar"></include>
```

El @layout/toolbar se pone así porque el layout que estoy importando se llama toolbar.

Para crear la flecha "atras" de la toolbar se hace así:

```
//con esto se crea una flecha hacia atras
Toolbar toolbar=findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

//Con esto creo un metodo para los click de la toolbar
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);
toolbar.setNavigationOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //con esto regreso a la pantalla anterior
        finish();
    }
});
```

para que se vea el toolbar con el menu de los 3 puntos se hace así:

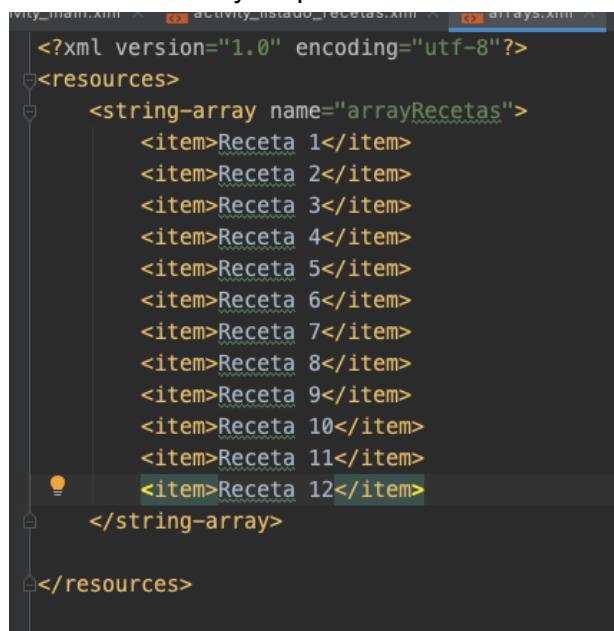
```
Toolbar toolbar = findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
```

LA TOOLBAR ES MUY IMPORTANTE, es *necesario* sustituirla por la actionbar (quitar la action y poner la tool).

Para sacar un listado de cosas usaremos el elemento listview. Para cargar cosas dentro del listview tengo que hacerlo desde su atributo “entries” y puedo cargar un array que yo quiera.

Vamos a crear un array de xml para llenarlo de cosas. Se hace en values -> new -> values resource

Llenamos un array de prueba:



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="arrayRecetas">
        <item>Receta 1</item>
        <item>Receta 2</item>
        <item>Receta 3</item>
        <item>Receta 4</item>
        <item>Receta 5</item>
        <item>Receta 6</item>
        <item>Receta 7</item>
        <item>Receta 8</item>
        <item>Receta 9</item>
        <item>Receta 10</item>
        <item>Receta 11</item>
        <item>Receta 12</item>
    </string-array>
</resources>
```

Para conseguir que este string-array se vea como un array en el arraylist hay que hacer lo siguiente dentro del protected void:

```
//declaramos el listview
ListView listaRecetas=findViewById(R.id.listrecetas);
//declaramos el adapter con el array convertido en values
ArrayAdapter<CharSequence> adaptador = ArrayAdapter.createFromResource(
    context: this,R.array.arrayRecetas, android.R.layout.simple_expandable_list_item_1);
//cargamos el adaptador de nuestro array.
listaRecetas.setAdapter(adaptador);
```

El createFromResource tiene 3 parametros: el primero es el contexto desde donde se llama a este array, el segundo es de donde va a coger el array para mostrarlo y el tercero es como y donde quieres que se muestre ese array.

A la hora de crear la app de recetas tengo que crear la clase receta para crear los objetos:

ATRIBUTOS

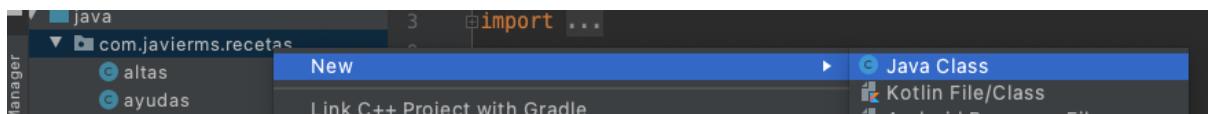
- idReceta:String
- nombre:String

- foto:int (las fotos las guardamos en un int en caso de que la guardemos en la carpeta drawable, una ruta si es de internet o un string en algun otro caso).
- descripcion:String
- valoracion:int
- idCategoria:int

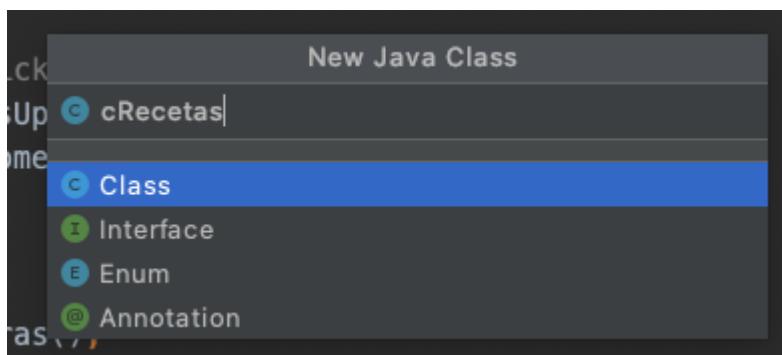
METODOS

- create()
- create(*argumentos*)
- todos los get() y todos los set() de los atributos

Ahora vamos a crear un archivo creador de objeto. Es un metodo que no va a estar vinculado a ninguna activity concreta, sino que sera global.



y luego



09/03/2021

Hoy haremos el login de la app. Usaremos firebase y hablaremos de seguridad.

```
//con esto lo que hago es poner la imagen de portada en el textView
ImageView imgPortada = (ImageView) findViewById(R.id.imgportada);
imgPortada.setImageDrawable(getDrawable(R.drawable.portadareceta));
```

El bitmap es lo que vamos a usar para redondear una imagen. Bitmap funciona por capas, por lo que nos sirve para modificar imagenes. Sirve tambien porque si tenemos muchas imagenes es mejor usar bitmap:

- si son pocas imagenes se puede hacer com photoshop.
- si son muchas imagenes, podemos usar el bitmap y un bucle para que se aplique a todas las imagenes que queramos.

```

// aqui hacemos un bitmap para que la imagen funcione por capas
// con el objetivo de hacer la imagen redondeada.
Drawable originalDrawable=getResources().getDrawable(R.drawable.portadareceta);
Bitmap imagenRedondeada=((BitmapDrawable)originalDrawable).getBitmap();

// creamos el drawable rodeon deado
RoundedBitmapDrawable roundedDrawable= RoundedBitmapDrawableFactory.create(getResources(), imagenRedondeada);
roundedDrawable.setCornerRadius(1000);

```

Aqui vemos como hacer un login sencillito pero que no tiene uso real

```

// con este codigo lo que hacemos es hacer una validacion de usuario/contraseña
// este metodo es muy maricarmen, el metodo bueno va a ser con firebase y con APIs
public void entrar(View view){

    EditText usuario = findViewById(R.idetxtusuario);
    String txtusuario = usuario.getText().toString();

    EditText contraseña = findViewById(R.idetxtpassword);
    String txtcontraseña = contraseña.getText().toString();

    if(txtusuario.equals("demo@demo.com") && txtcontraseña.equals("demo")){
        Intent intent = new Intent(getApplicationContext(), listadoRecetas.class);
        startActivity(intent);
    } else {
        Toast.makeText(getApplicationContext(), text: "usuario o contraseña incorrectos", Toast.LENGTH_SHORT).show();
    }
}

```

un bucle if dentro de un metodo hace el recorrido hasta que llega a un return. Este caso, por ejemplo:

```

public boolean isValido(String user, String pass){
    if(user.equals("demo@demo.com")&&pass.equals("damo")){
        return true;
    }
    return false;
}

```

como este metodo me tiene que devolver un boolean, el metodo deja de ejecutarse en el momento en que reciba un return. En este caso, se hace la comprobacion del if, y si es true el metodo devuelve true y deja de ejecutarse, y si es false, el metodo no entra en el bucle y se ejecuta directamente el false.

Empezamos con firebase.

Hablamos de firebase porque estamos viendo la autenticacion de la app. Cuando me autentico en algun sitio lo que hago es establecer un puente hasta una API y para ello necesito internet, pero si no hay conexion voy a tener un problema. Tengo que ver que cosas me interesa tener en la nube y cuales en el dispositivo para que no dependa de internet 100%.

Despues de instalar firebase tenemos que poner el android studio en modo debug, por que sino, no nos dejara trabajar.

11/03/2021

Para que se vea el menu de la toolbar hay que ponerle un color. Para ello, hacemos el item en la parte de themes.

```
<!-- Status bar color. -->
<item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
<item name="android:textColor">#000</item>
```

El titulo que aparece en el toolbar no puedo modificarlo, pero lo que si que puedo hacer es ponerle un textview encima y modificar el texto del textview. Para ello, el text view tiene que estar dentro del toolbar y tabulado y dentro del toolbar (fijate que el toolbar se abre al principio pero la etiqueta de cerrado se pone mucho mas abajo).

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr actionBarTheme"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:title="RECETAS">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/app_name"
        android:textSize="24sp"
        android:textColor="@color/white"
        android:textAlignment="center"/>

</androidx.appcompat.widget.Toolbar>
```

En la activity de altas los menus desplegables se llaman spinner.

Para acceder a algun servicio del telefono como la galeria o la camara, lo tenemos que hacer desde el android manifest. Tenemos que poner ahí todos los servicios que queremos que tenga nuestra app. Con estas dos lineas lo que hago es pedir permiso para coger datos del telefono.

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Una vez que tenemos los permisos, tenemos que vincular las cosas.

Para coger la galeria necesito un metodo pero ese metodo NO puede ser con un boton, no puedo hacerlo con un view. Puedo hacer un view para un boton y que al pulsar el boton me llame al metodo que cargue la galeria, pero no puedo cargar la galeria directamente.

```
// con este metodo lo que hago es hacer cosas con el boton
public void cargaFotoGaleria(View view){
    cargarGaleria();
}

// con esto lo que hago es acceder a la galeria de forma segura
// es un metodo aparte porque no puedo acceder directamente a la galeria
// tengo que hacer un metodo que invoque a este metodo para llegar a la galeria
// y el requestCode 10 nos permite seleccionar la aplicacion que gestione la galeria.
private void cargarGaleria(){
    Intent intent = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(intent, requestCode: 10);
}
```

```
//con este metodo lo que hago es recoger la foto seleccionada en la galeria
// y la guardo en el imageView que he declarado como variable global
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode==RESULT_OK){
        Uri path =data.getData();
        imgCarga.setImageURI(path);
    }
}
```

Cuando seleccionas la imagen y la pones como imageView NO LA ESTAS SUBIENDO A NINGUN SITIO, solo la estás recogiendo.

Para guardar fotos vamos a tener que mirar la API en la pagina de Android Developer. Como las camaras tambien las mejoran mucho, pues los metodos que interactuan con ellas pues hay que actualizarlos. Lo que hay que hacer es irse a la pagina de Android Developer, mirar la documentacion poniendo "camera" en el buscador e implementar lo que toque.

15/03/2021

Cuando creamos una clase tiene que tener una apertura de clase y un metodo constructor. Un metodo constructor tiene que tener el mismo nombre que la clase sobre la que esta construyendo, es decir, si la clase se llama cRecetas el constructor se tiene que llamar cRecetas y no se usa el parametro de retorno (no se le pone void ni nada).

```
//metodo constructor del objeto
public cRecetas(int id, String nombre, String idCategoria, String descripcion, int foto, int valoracion){
    this.cRidReceta=id;
    this.cRidCategoria = idCategoria;
    this.cRdescripcion=descripcion;
    this.cNombreReceta=nombre;
    this.cRfoto=foto;
    this.cRvaloracion=valoracion;
}
```

Para añadir un objeto a la clase primero tenemos que tener un constructor. Luego necesitamos meter los datos y coger los datos para grabarlos en la clase.

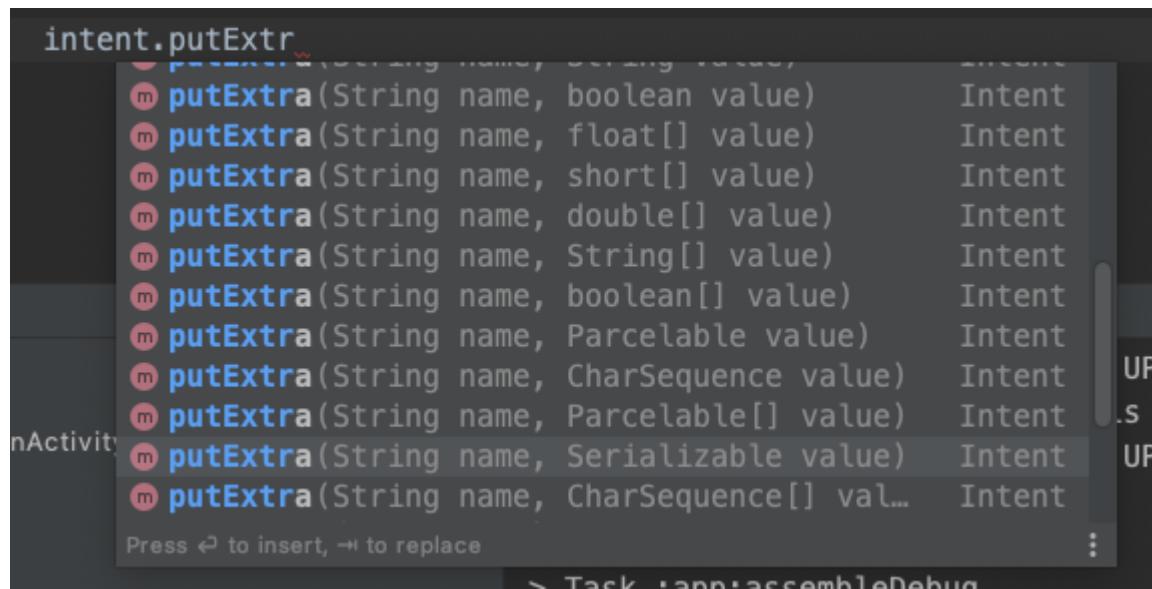
Ahora vamos a ver como pasar objetos de una pantalla a otra. Para ello, los pasos son:

- Intent: this, activity.class
- Añadir intent objeto -> intent.putExtra(String, serializable). Muy muy importante el poner que el objeto sea serializable.
- startActivityForResult(intent).

Para que nos funcione bien el intent, en la clase recetas tenemos que poner implements Serializable.

```
//clase que implementa el serializable
public class cRecetas implements Serializable {  
    //atributos
```

Cuando quiero enviar un objeto a otro sitio tengo que hacer un Intent y luego ponerle un putExtra(serializable).



Así enviamos el objeto.

```
public void guardar(View view){
    /*Lo que hay que hacer aquí es:
     - Recoger los elementos de pantalla
     - Guardar to-do en el objeto receta.
     - Objeto o campos enviarlo por un Bundle a la pantalla de detalle (para que quede claro que se ha guardado) y luego a lista.
     */

    // recoger los elementos de la pantalla
    EditText txtNombre=findViewById(R.id.editNombreReceta);
    EditText txtDescripcion=findViewById(R.id.editDescripcion);
    RatingBar rtbValoracion=findViewById(R.id.starValoracion);

    receta.setcRnombreReceta(txtNombre.getText().toString());
    receta.setcRdescripcion(txtDescripcion.getText().toString());
    receta.setcRidCategoria(listaCategorias.getSelectedItem().toString());
    receta.setcRvaloracion(rtbValoracion.getRating());
    //receta.setcRfoto(imgCarga);

    Toast.makeText(getApplicationContext(),
        text: "Nombre: "+receta.getcRnombreReceta()
            + " Descripcion: "+receta.getcRdescripcion()
            + "Categoria: "+ receta.getcRidCategoria()
            + "Valoracion: "+receta.getcRvaloracion(), Toast.LENGTH_LONG).show();

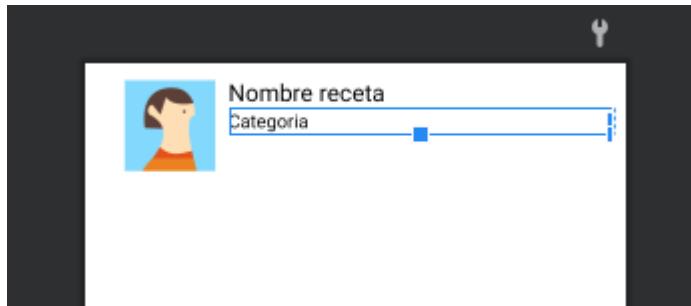
    Intent intent = new Intent(getApplicationContext(), detalle.class);
    intent.putExtra( name: "Receta", receta);
    startActivity(intent);
}
```

Ahora lo recogemos en la pantalla de detalle. Para ello, lo que hacemos es:

- Crear el objeto.
- Llamamos al Intent
 `getIntent().getSerializableExtra("aqui va un identificador (key) en string");`

Para que en el listview aparezcan los objetos que hemos creado salgan con una imagen de lista y con una pequeña imagen, el nombre y la categoria. Para ello, tenemos que crear un

itemlist, que es un layout nuevo que tiene únicamente lo que queremos mostrar, y luego haremos un bucle para que se rellene y se muestre.



Siempre que creamos un array en android usaremos arrayList para luego enseñarlo en el listado de recetas. Para declarar el arrayList y decirle que lo que queremos es un arrayList de los objetos de cRecetas se escribe así:

```
// declarar un array de objetos de recetas
ArrayList<cRecetas> arrayList =new ArrayList<cRecetas>();
```

Para que el arrayList funcione tenemos que hacer un adaptador customizado, un custom element. Para crear el adaptador, tenemos que crear una nueva clase de Java, la llamaremos RecetaAdapter y la empezamos a llenar. Hay dos parametros que no se me tienen que olvidar:

- el contexto: donde funciona.
- el recurso: que tipo de recurso es el que esta usando.

```
public class RecetaAdapter extends ArrayAdapter<cRecetas> {

    private Context miContexto;
    private int miRecurso;

    public RecetaAdapter(@NonNull Context context, int resource, @NonNull ArrayList<cRecetas> objects) {
        super(context, resource, objects);
        this.miContexto = context;
        this.miRecurso = resource;
    }
}
```

IMPORTANTE

Con este código lo que consigo es pasar de la lista de todas las recetas al detalle de una receta en concreto. Importante: por temas de memoria y rendimiento SOLO QUIERO QUE ME MUESTRE LA RECETA QUE LE HE PEDIDO, no quiero que me mande todo el array de todas las recetas.

```

//cuando se pulsa un item del listview
listaRecetas.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

        Intent haciaDetalle=new Intent(getApplicationContext(),detalle.class);

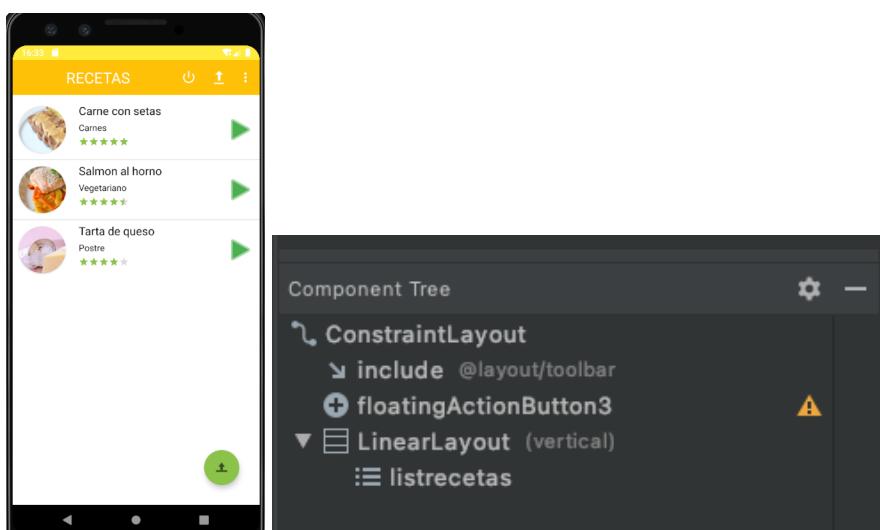
        haciaDetalle.putExtra( name: "receta", arrayList.get(position));

        //aqui voy a la actividad que marca el intent
        startActivity(haciaDetalle);
    }
});

```

16/03/2021

Para añadir un botón flotante el elemento se llama floating action button. Es un dolor de huevos ponerlo para que se vea bien. El código para que se vea así es:



```

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/floatingActionButton3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="32dp"
    android:layout_marginBottom="32dp"
    android:clickable="true"
    android:onClick="altaReceta"
    app:layout_constraintBottom_toBottomOf="@+id/include"
    app:layout_constraintEnd_toEndOf="parent"
    app:srcCompat="@android:drawable/ic_menu_upload" />

```

Para cambiar el color de un ícono primero tengo que localizar el atributo `srcCompat` y borrarle el `Compat` para que se quede `src` solo, luego en `backgroundTintMode` puedo usar `add` o `multiply`:

- `add`: trabaja sobre el propio ícono.

- multiply: trabaja sobre el src que he puesto.

Ahora quiero guardar la receta en el arraylist. Con este código es con lo que creo nuevos elementos, pero cuidado: este elemento no se queda en la memoria, por lo que el primer elemento se crea pero el segundo elemento lo que hace es sustituir al primero.

```
// aquí recojo el objeto que viene de la activity altas.
cRecetas recetaAlta = (cRecetas) getIntent().getSerializableExtra( name: "receta");

if(recetaAlta!=null) {
    int indice = arrayList.size();
    indice++;
    arrayList.add(new cRecetas(indice,
        recetaAlta.getcRnombreReceta(),
        recetaAlta.getcRidCategoria(),
        recetaAlta.getcRdescripcion(),
        R.drawable.recetatartaqueso,
        recetaAlta.getcRvaloracion()));
}
```

Cuando utilice los códigos de la API de cámara tengo que tener cuidado con el manifest porque en la página de android me pone que ponga

```
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="com.example.android.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths"></meta-data>
</provider>
```

Y lo que tengo que poner es

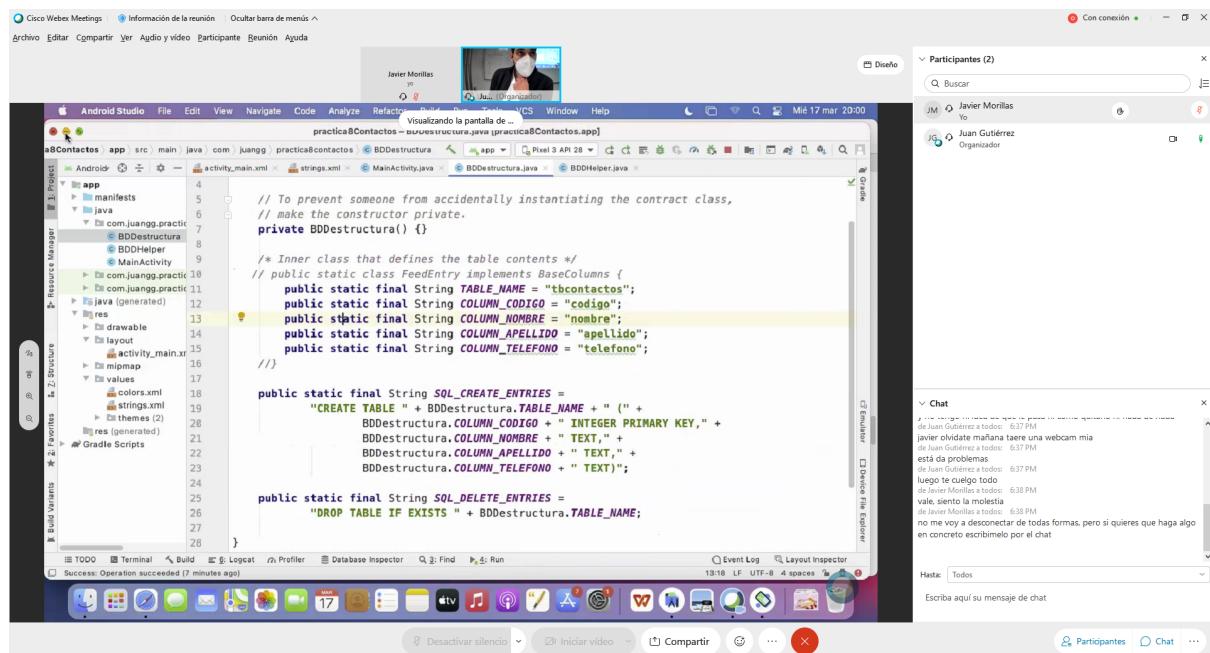
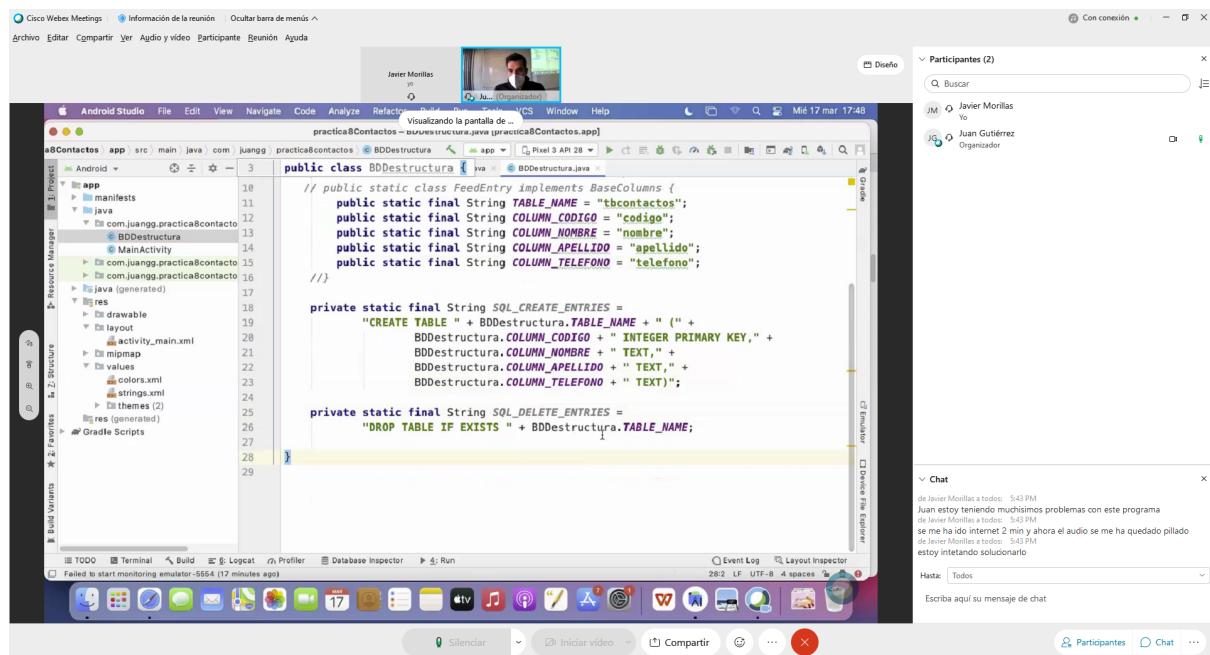
```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="com.example.android.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths"></meta-data>
</provider>
```

IMPORTANTE

Cuando queremos subir una app a google play necesitamos que la APK este en modo release, no en modo debug. Para crear una APK en modo release tenemos que hacer lo siguiente:

esto tengo que mirarlo

17/03/2021



22/03/2021

Hoy empezamos con los fragments. Los fragments lo que me permiten es repetir los fragments en distintas activities, como por ejemplo un menu. Yo hago un fragment de un menu y luego puedo hacer que ese fragment aparezca en todas las activities.

para crear un fragment hay que darle app->new->fragment -> fragment (blank)

los fragments se tienen que comunicar con las activities, y eso lo hacen con los interfaces. Para crear una interfaz hay que darle a app->java class y luego ahí dentro le pongo nombre y en lugar de class le tengo que poner interface.

23/03/2021

Para pasar un fragment a otro activity tengo que usar la etiqueta <fragment> y dentro de esa etiqueta tengo que usar el valor name="" y ponerle el path, por ejemplo:

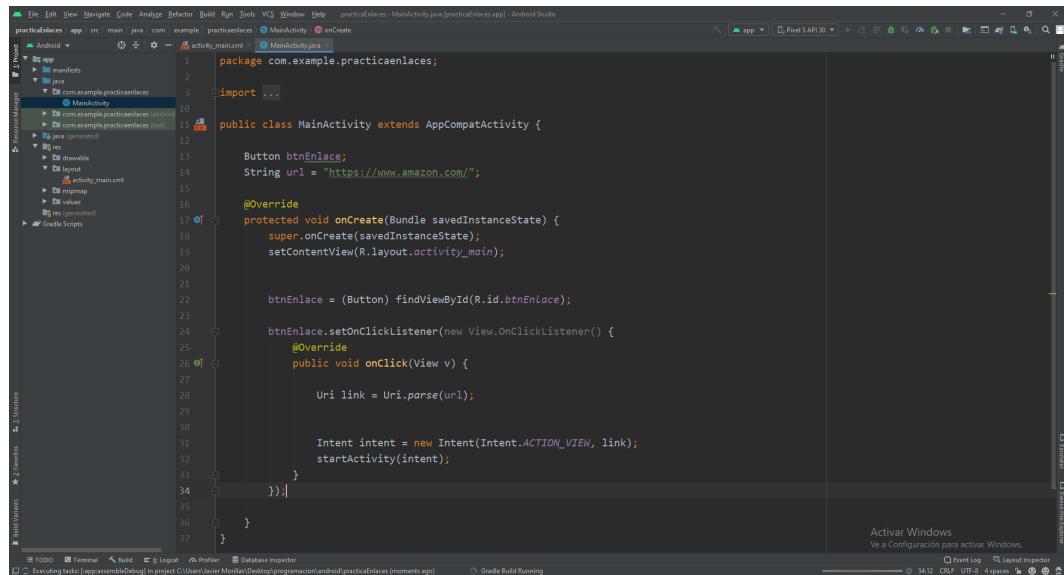
```
<fragment
    android:id="@+id/menu"
    android:name="com.example.practicafragments.menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
/>
```

IMPORTANTE

en los fragments NO SE PONE getApplicationContext cuando haces un Intent, lo que pones es getActivity, porque no sabes desde donde se llama a esa activity.

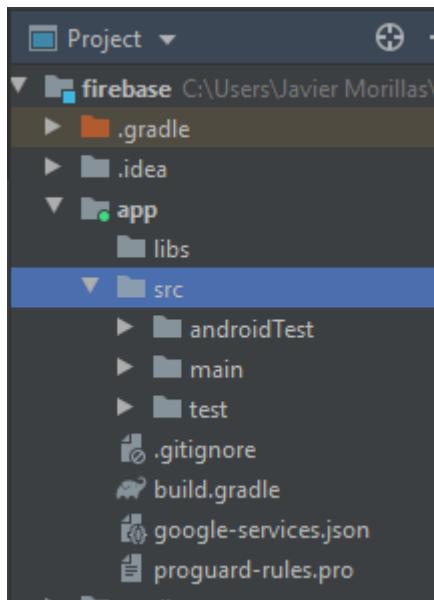
25/03/2021

Para poner un link a un sitio creamos un botón y le ponemos a escuchar:



Ahi lo que he hecho ha sido poner un boton que al pulsarlo me lleve a amazon. Cuando quiero poner un link tiene que ser en formato URI, por eso hago primero un String y luego lo paso a Uri.

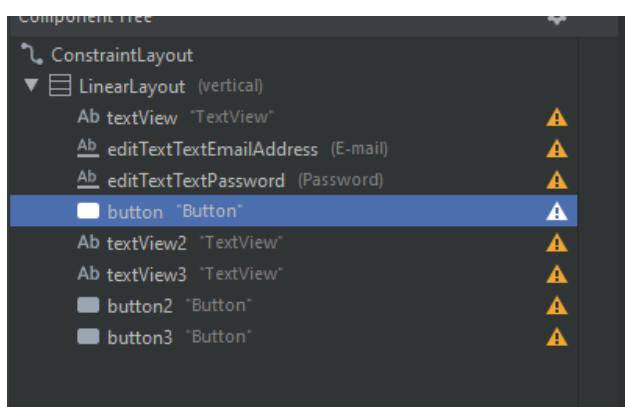
Estamos con firebase. Para instalar el archivo google-service.json tenemos que poner la app en modo proyecto (donde pone app justo debajo de file).

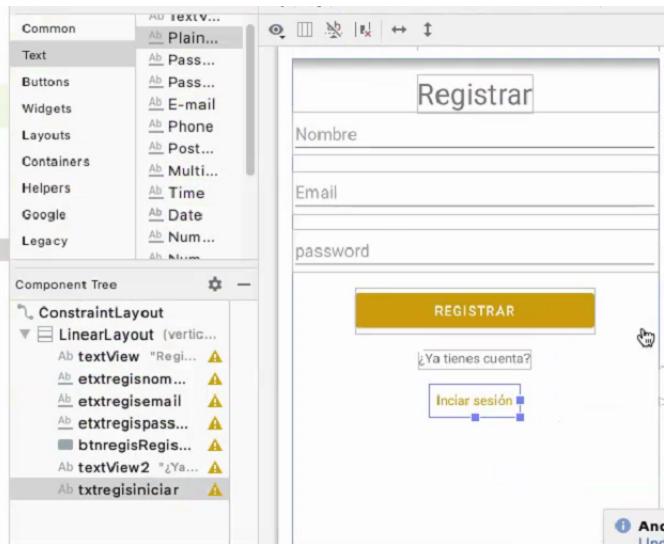


26/03/2021

Vamos a hacer autenticación con firebase.

Primero creamos el layout con esto y le ponemos bien los id:





Luego en el main activity declaramos todas las cosas:

```
public class MainActivity extends AppCompatActivity {

    EditText etxtemail, etxtpassword;
    Button btnEntrar, btnGoogle, btnFacebook;

    TextView btnRegistrar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

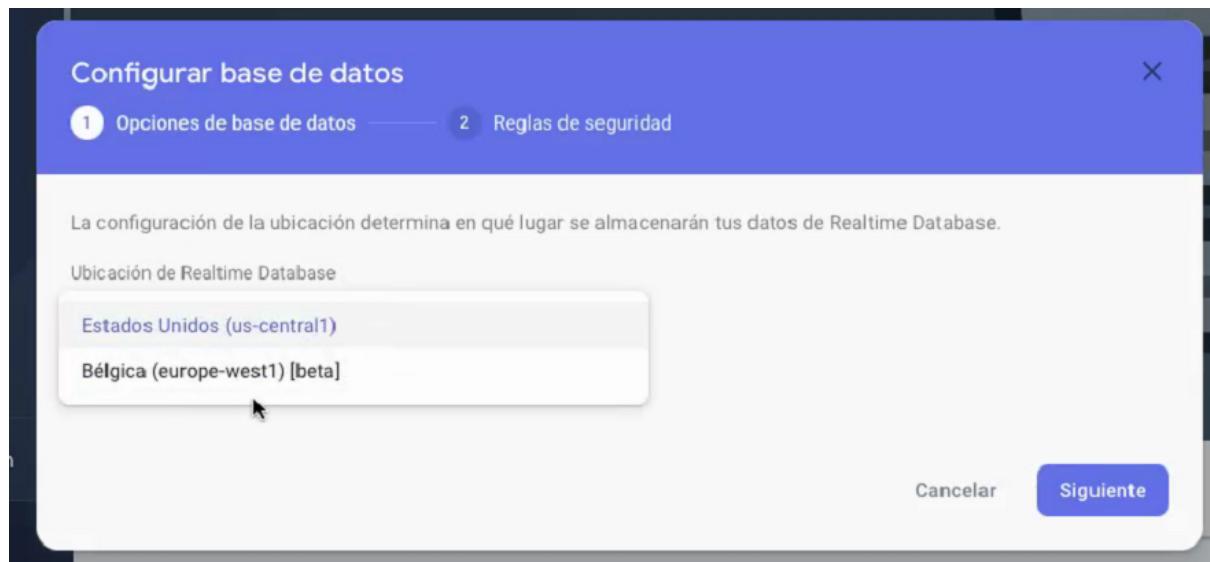
        etxtemail = (EditText) findViewById(R.id.etxtacessemail);
        etxtpassword = (EditText) findViewById(R.idetxtacesspassword);

        btnRegistrar = (TextView) findViewById(R.id.txtacessregistrar);
        btnEntrar = (Button) findViewById(R.id.btnEntrar);
    }
}
```

el btnRegistrar le ponemos a escuchar con un clickListener.

Si en algun momento tengo que poner configurar la base de datos de firebase tengo que elegir entre EEUU o Belgica:

- normalmente, EEUU funciona mejor pero:
- Si mi empresa va a usar firebase y la tengo tributando o localizada en españa, tengo que usar la base de Belgica por ser Europa.



Tengo que hacer un metodo para que compruebe si los campos de usuario, mail y registro estan vacios (es un con un if y el metodo isEmpty, lo que se escribiria asi !nombre.isEmpty para comprobar si el campo "nombre" está vacío.). Si ninguno de los parametros esta vacio, entonces le dire que haga el metodo registrar usuario.

```
private void registrarUsuario() {  
  
    //comprobar si es un usuario valido para autenticacion  
    mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
        @Override  
        public void onComplete(@NonNull Task<AuthResult> task) {  
            if(task.isSuccessful()){  
  
            } else {  
                Toast.makeText(getApplicationContext(), text: "No se ha podido registrar", Toast.LENGTH_LONG).show();  
            }  
        }  
  
    }  
  
    private void registrarUsuario() {  
  
        //comprobar si es un usuario valido para autenticacion  
        mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>()  
            @Override  
            public void onComplete(@NonNull Task<AuthResult> task) {  
                if(task.isSuccessful()){  
  
                    //crea un id para el usuario de tipo alfanumerico y único  
                    String id = mAuth.getCurrentUser().getUid();  
  
                    mDatabase.child("Users").child(id).  
  
                } else {  
                    Toast.makeText(getApplicationContext(), text: "No se ha podido registrar", Toast.LENGTH_LONG).show();  
                }  
            }  
        }  
    };
```

Android Studio 4.1.3 available
Update...

5/04/2021

Hoy hablamos de los estados de las activitys. Estan en la presentacion numero 23 en el servidor. Este tema es importante porque los estados de las activities es super IMPORTANTE, lo preguntan mucho en los examenes de certificacion.

- el metodo onCreate es OBLIGATORIO para que la app funcione. Sin el onCreate, la app no funciona.
- el metodo onStart lo que hace es que la activity sea visible. Primero se hace el onCreate y luego el onStart para que la activity sea visible y se muestre por pantalla.
- el metodo onResume es llamado despues del onStart. Cuando la actividad ya esta visible despues del onStart, se va a poder interactuar con ella con el metodo onResume.
- el metodo onPause se produce cuando la app pierde el foco, es decir, sale de pantalla pero no se apaga (basicamente, es dejarla en segundo plano, que no esta abierta pero no esta cerrada). Cuando la app esta en onPause pueden pasar 2 cosas: que la app ejecute el metodo onResume o el metodo onStop.
- el metodo onStop lo que hace es que la activity deje de ser visible.
- onRestart
- onDestroy

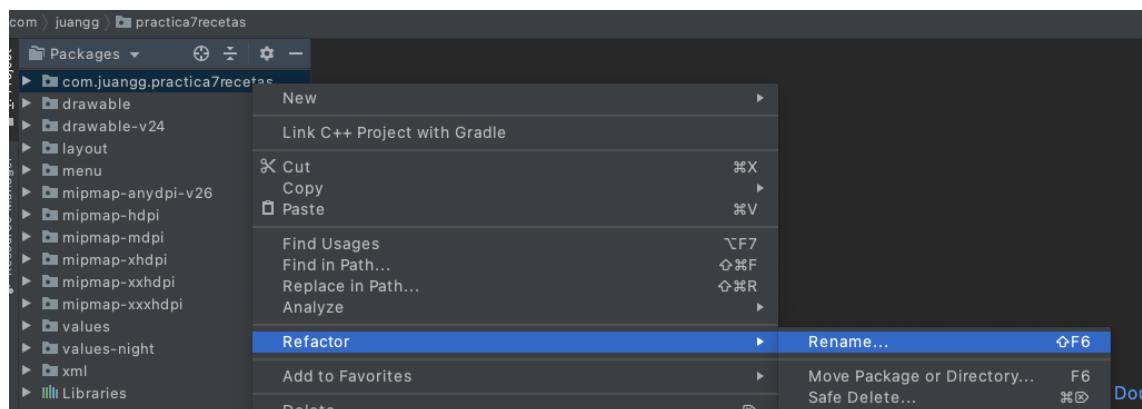
06/04/2021

Estudiar lo que seria un examen de certificacion de android. Estudiar los estados de la app (lo que esta justo arriba y las presentaciones del drive).

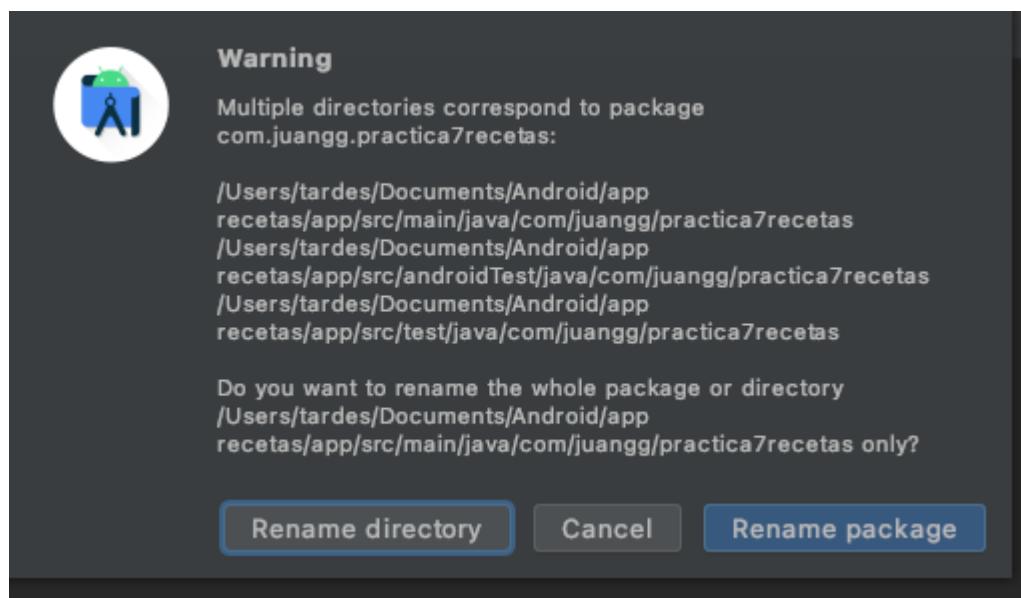
07/04/2021

estudiar certificacion de android.

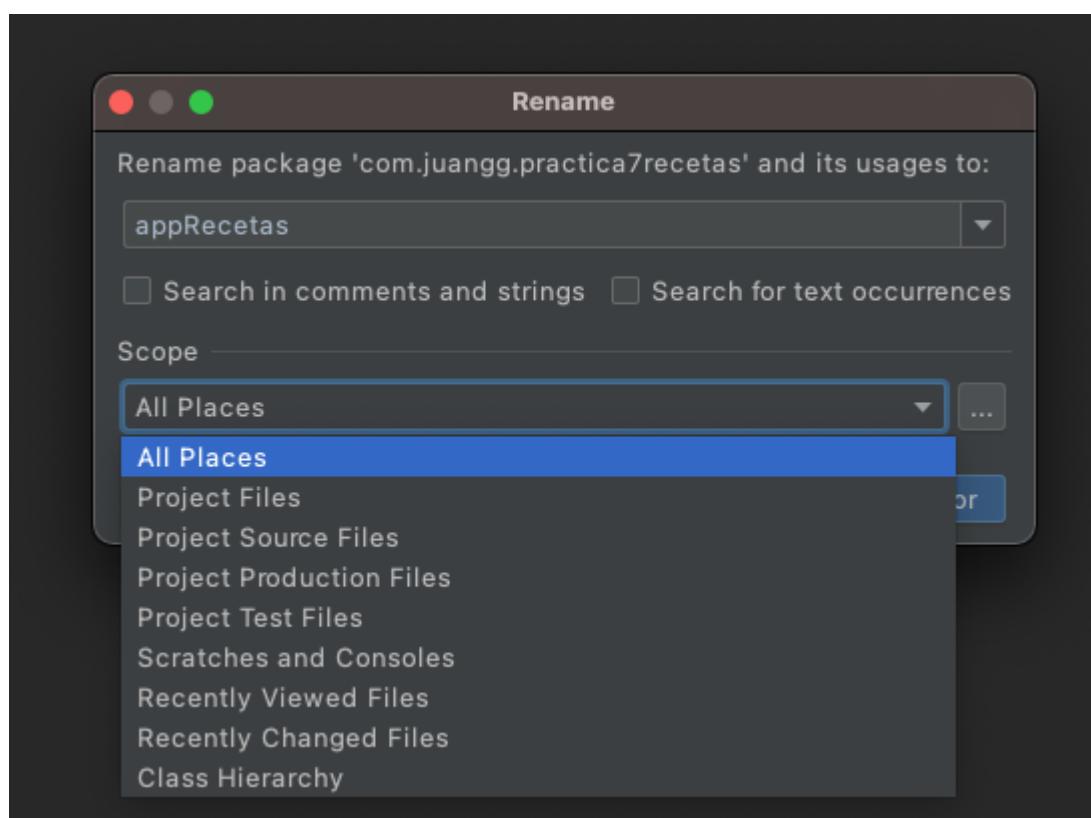
Para cambiar el nombre a todo el proyecto hay que darle a package (debajo de file) y hacerle refractos->rename.



Y aqui hay que darle a rename package



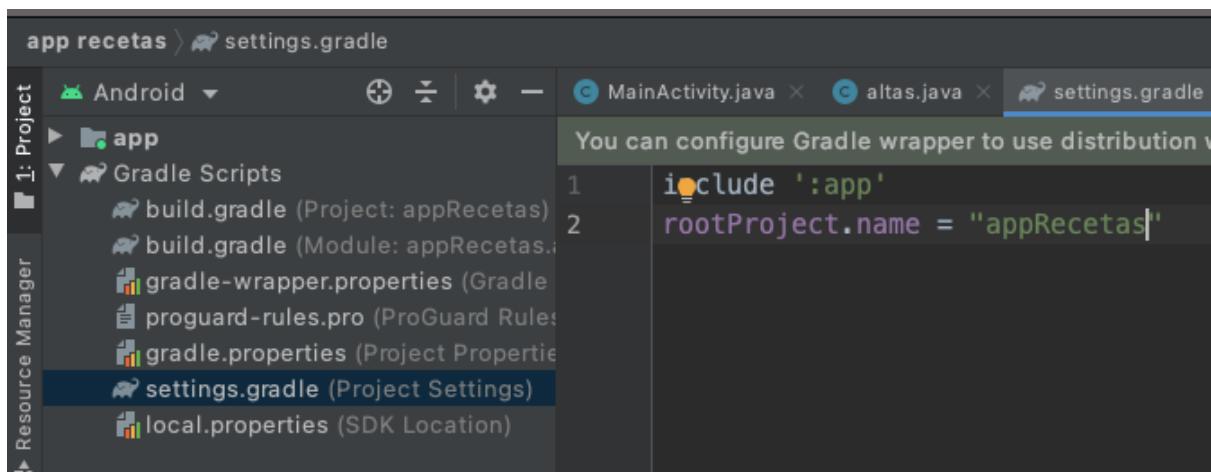
Y aqui hay que darle al nombre que quieras y en scope hay que poner "all places"



Si hay que seguir refactorizando para que me siga cambiando nombres dentro de la app, tengo que darle a file->project structure->modules, y ahí cambiar el nombre

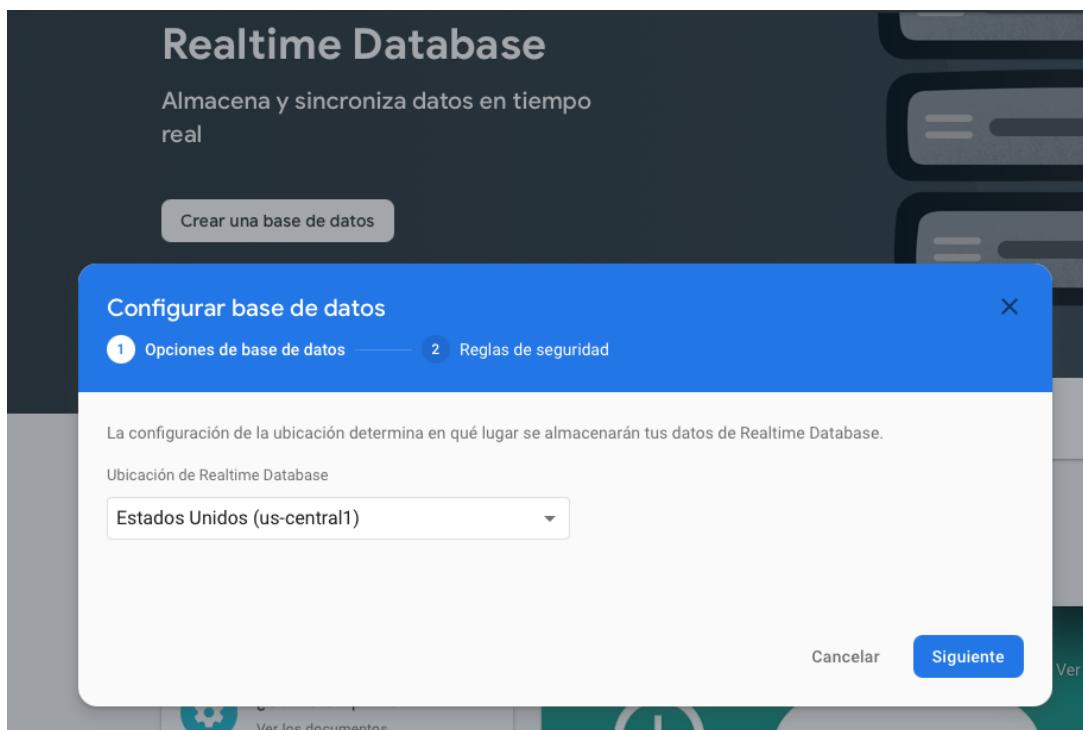


Si sigue dando fallo, podemos mirar aqui tambien



Ahora usaremos firebase como base de datos para guardar y recuperar datos. Lo primero que hay que hacer es conectar con firebase y vincularle la app correspondiente.

Luego tengo que crear la base de datos y ponerle que este en EEUU



Código importante para subir y recoger datos de la base de datos.

```
miDataBase = FirebaseDatabase.getInstance();
DatabaseReference miReferencia = miDataBase.getReference( path: "mensaje");
// esto es para escribir en la base de datos
miReferencia.setValue("Hola, ya tenemos base de datos.");
//esto es para leer la base de datos
miReferencia.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String mensaje = snapshot.child("mensaje").getValue().toString();
        TextView txtmensaje = findViewById(R.id.txtmensaje);
        txtmensaje.setText(mensaje);
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
```

Cuando creamos hijos en una base de datos es conveniente que creemos un hijo del hijo, de forma que el id no sea igual al valor de la categoría, sino que el id sea el padre del valor de la categoría. De esa forma, consigo que la base de datos se me quede “abierta” y pueda ampliarla sin problemas.

```
btnAltaCategoria.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String comprobarCategoria = txtCrearCategoria.getText().toString();
        if(!comprobarCategoria.equals("")){
            miReferencia.child("categoria").push().child("nombreCategoria").setValue(comprobarCategoria);
            finish();
        } else {
            Toast.makeText(getApplicationContext(), text: "Introduce una categoría válida.", Toast.LENGTH_LONG).show();
        }
    }
});
```

Para guardar los datos de forma ordenada y bien tengo que usar el map (que no se que es y tengo que mirar como se usa).

```
btnAltaCategoria.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Map<String, Object> Map;
        HashMap<String, Object> mapa = new HashMap<>();
        mapa.put("id","0");
        mapa.put("ncategoria",txtCrearCategoria.getText().toString());
        mapa.put("rutafoto","/rutafoto");

        miReferencia.child("categoria").push().setValue(mapa);
```

08/04/2021

Empezamos hablando de firebase.

Cuando trabajamos con firebase necesitamos primero hacerle una referencia, para ello hacemos un DatabaseReference (lo que nosotros solemos asignar a una variable que llamamos miReferencia).

Tambien es importante usar el dataSnapshot (mirar la documentacion).

Cuando queremos acceder a algún nodo concreto de la base de datos tenemos que usar la palabra child, por ejemplo miReferencia.child("el nombre del nodo al que queremos acceder").

Las ordenes importantes para gestionar una base de datos son escribir (setValue), leer (getValue) y eliminar (removeValue).

Si por ejemplo queremos acceder a una parte de la base de datos que es muy compleja (que tiene muchos subnodos) hay que ponerle .child por cada nodo que exploramos. Por ejemplo:

- apprecetas
 - categorias
 - nombre

Aqui se accederia a nombre mediante

miReferencia.child("apprecetas").child("categorias").child("nombre")

Para recorrer una base de datos para hacer una comprobacion en Android tenemos que usar dataSnapshot en lugar de con un bucle directamente.

IMPORTANTE: cuando creo un objeto para que me lo muestre en la app cogiendo datos de firebase necesito un metodo constructor vacio, totalmente vacio, porque el programa crea el objeto entero de nuevo con ese metodo constructor pero lo tiene que crear de 0.

```
//Añadir un item receta de la base de datos
miReferencia.child("recetas").child("el codigo que sea").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if(snapshot.exists()){
            //si encuentra la receta
            cRecetas unaReceta = snapshot.getValue(cRecetas.class);
            arrayList.add(unaReceta);
        } else {
            //si no la encuentra
        }
    }
})
```

```
public cRecetas(){  
}
```

Para guardar datos:

- primero inicializamos firebase con instancia y referencia.

```
public class listado extends AppCompatActivity {  
  
    ListView listRecetas;  
  
    private FirebaseDatabase miDatabase;  
    private DatabaseReference miReferencia;  
  
    ArrayList<cRecetas> arrayList;  
    RecetaAdapter recetaAdapter;
```

- luego uso el metodo inicializarFirebase que tiene 3 ordenes (coger instancia, coger referencia y empezar a usar google services y analytics).

```
private void inicializarFirebase() {  
    //inicializa la instancia de firebase app con valores de recursos de google-service.json y tambien inicializa firebase analytics  
    FirebaseApp.initializeApp(this);  
    //lineas de la base de datos  
    miDatabase = FirebaseDatabase.getInstance();  
    miReferencia = miDatabase.getReference();  
}
```

09/04/2021

Hoy hablamos de comercializacion de apps.

Cuando hago una app tengo que preguntarme si quiero hacerla nativa o no nativa.

- nativa: estar hecho en Java, Kotlin, Objective C o Swift
- no nativa: esto es mas una web app que lleva una serie de frameworks.

La diferencia entre ellas es:

- las apps nativas es mas dificil hacerlas y es mas complejo.
- Las apps nativas tienen la ventaja del acceso a los componentes del telefono.
- Las apps nativas son mas rapidas a la hora de ejecutarse porque tienen acceso directo a la ram y al hardware del telefono.
- las apps no nativas es practicamente hacer una pagina web.

- Las apps no nativas no tienen un acceso tan sencillo a los componentes del telefono.

A nivel de precio Juan cobra de 8.000€ en adelante por cada app nativa que hace porque:

- tiene que hacer dos versiones de la app (una para android y otra para ios) (y por cada version cobra la app entera)
- por cada servicio o cosa que haga la app hay que sumarle 600€
- Esto son unos 6 meses de trabajo

Por cada app no nativa Juan cobra 3.500€:

- el mejor framework para trabajar es ionic porque sirve para android y para ios
- esto son unos 2 meses de trabajo.

Mirar empresa Ditrendia. Para ganarse la vida como programador de apps hay que trabajar por proyectos y puedes pillar proyectos por una agencia o puedes hacer apps tuyas propias como desarrollador.

Sobre la monetizacion:

- Se puede sacar dinero monetizando apps subiéndola a la store.
 - pueden ser:
 - gratuitas pero te genera dinero con la publicidad con los banner
 - de pago
 - mixtas: apps gratuitas que tienen ciertos modulos de pago.

Para mirar temas de monetizacion podemos mirar google ad sense (hay un componente en android que te permite conectar con googleAdSense directamente).

Cuando subimos una app a google play el éxito de la app depende de:

- descargas/compras en la app
- opiniones
- posicionamiento de la app en buscadores (que se traduce en las veces que aparece en los buscadores).

Lo suyo es que cada app que hagamos tenga una pagina web asociada.

Para registrar una marca o app se hace en la pagina de registros y marcas.

Al subir una app pues va a tener una mierda de opiniones y una mierda de descargas, así que hay que hacer algo para que se vea, por eso es importante hacer una web porque aumenta el SEO y el SEM.

- SEO: es la parte que se ocupa de la optimización de la visibilidad.

- SEM: es la parte que se ocupa del pago.

Ayudamos al SEO con:

- web
- landing
- redes
- linkbuilding

LO PRIMERO que hay que hacer es saber a quien quiero vender mi app, hacer un estudio de mercado para saber a que sector quiero venderle esa app.

Cuando creamos la web de la app tiene que ser simple, muy muy simple. Tiene que tener 2 elementos principales:

- que la web diga de que va mi app
- que la web diga por que es interesante bajarse mi app (ponerle links a sitios que generen interes en mi, como blogs, redes sociales...).

Hay una pagina que se llama ThemeForest que vende "moldes" para wordpress. Dentro de ThemeForest buscamos "landing app" y ahí buscamos la que nos guste. Juan usa una que se llama app landing page y cuesta 39\$. Para instalar eso y ponerla en nuestra landing page tendríamos que comprar un hosting y un dominio (Juan recomienda ionos y un hosting con wordpress). Si queremos instalar wordpress lo mejor es descargarlo desde wordpress.org y luego hacer el hosting donde queramos.

para saber que palabras usar para que el SEO se haga bien y eso se puede hacer con una cosa que se llama planificador de palabras. Juan recomienda Ubersuggest, que lo que hace es que tu le metes una palabra y la pagina te dice palabras y temas relacionados para que los usemos para el SEO. Cuando hagamos una web lo que tenemos que hacer es UNA SOLA WEB PARA CADA APP, y cada app SOLO RESUELVE 1 SOLA COSA, no un montón.

Trends24 es una pagina que me enseña los trendings topic de España y te enseña lo más hablado para ver si puedes sacar algo. Con google trends lo que puedes conseguir es saber cuánto se ha buscado una palabra.

Yoast SEO es un plugin para wordpress que es muy recomendable para posicionar bien.

Para posicionar bien la app:

- tengo una landing page que cuenta de que va la app pero tiene 2 botones principales: el de google play y el de app store. La landing page no tiene enlaces

externos, no quiero que la gente se vaya de mi pagina a menos que sea para bajarse la app.

- tengo tambien un blog que tiene entradas y noticias que explotan las palabras clave, son entradas que usen las keyword para que se posicionen y vendan la app.
- cada una de esas paginas tiene que tener su propio SEO, su propio posicionamiento.

EN WORDPRESS

para posicionar:

- me voy a ajustes generales y tengo que ponerle el titulo del sitio y la descripcion corta de la pagina con keywords.
- tengo que hacer que los motores de busqueda me encuentren. Eso lo pongo en ajustes -> lectura, y le quito el tick de desuadir a los motores de busqueda.
- tengo que instalar un pluggin que se llama yoast SEO y el pluggin lo que nos hara es contarnos que tal va todo el posicionamiento.
- nuestro mapa del sitio es bien sencillo: pagina principal, pagina general del blog y cada entrada del blog.
- para posicionar una entrada tengo que:
 - escribir un titulo que este relacionado con la URL de la pagina y tiene que aparecer la palabra clave que yo tenga. Por ejemplo, si mi entrada es "como jugar al ajedrez" la URL sera <http://blablabla.com/como-jugar-al-ajedrez>
 - antes de publicar la entrada, bajar en la pagina y mirar el yoast SEO y aplicar todo lo que me diga.
 - poner la densidad de la keyword adecuada, por lo menos una vez por cada 2 parrafos.
 - tengo que dar una meta descripcion, tengo que hacer una descripcion para que google la indexe en una especie de preview de la pagina.
 - super importante que si meto una imagen, el nombre de la imagen tenga alguna keyword.
 - registrar mi pagina en buscadores, por ejemplo en google search console y sirve para indexar la web. En la pagina le doy a añadir propiedad y le pongo mi dominio ahí.
- Tengo que enlazar tanto mi app como mi web con google analytics.
- para crear trafico hacia mi web y conseguir posicionarla tambien puedo usar google ads. Google ads funciona como una especie de subastas para ganar las primeras posiciones en el buscador de google. Promocionar tu pagina en las redes es poco efectiva y barata, google ads es mejor pero muchisimo mas cara.
- Woocommerce pluggin para wordpress es con lo que se puede hacer una pasarela de pagos.

12/04/2021

Hoy hablamos de comercializacion de apps.

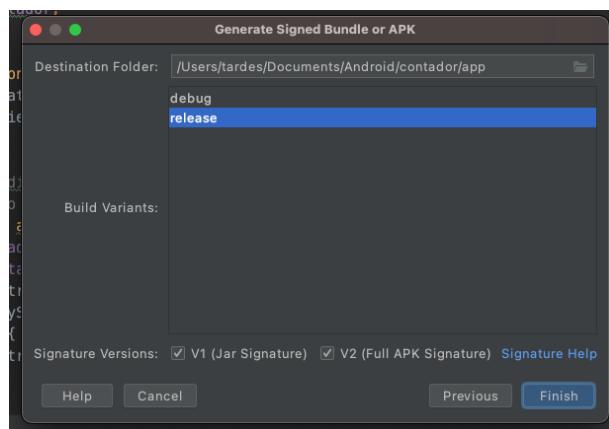
Lo primero que hay que hacer a la hora de comercializar la app es registrar la marca o branding.

SEO - Hacer todo lo que pueda para que mi web este bien optimizada y que aparezca facilmente en los buscadores utilizando las keywords.

En ubersuggest tengo la opcion de coger keywords relacionadas con lo que quiero empezar, hay un apartado de sugerencias y de preguntas. Lo mejor es usar las palabras clave de sugerencias para las paginas principales pero en un blog o en algo que quiera para captar mas atencion, usar las palabras de las preguntas.

En un wordpress para hacerle el SEO bien tengo que poner en ajustes generales el titulo del sitio, la descripcion corta; luego en lectura tengo que deshabilitar el tick en visibilidad en los motores de busqueda; instalar el pluggin yoast seo.

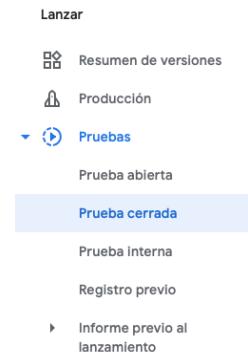
IMPORTANTE: cuando subimos una app a google play tenemos que subir la apk en modo release, no en modo debug. En la parte de signature versions tenemos que darle a los dos a V1 y V2.



En el panel de control de la play store aqui es donde voy a modificar las cosas del SEO, aqui es donde pongo el nombre y las descripciones. Tambien podre modificar absolutamente todo lo que tenga que ver con la app.



Y para hacer pruebas lo mejor es hacer una prueba cerrada, que es como mandar una invitacion para que la gente lo pruebe.



Si tengo algún problema a la hora de subir la apk a la play store tengo que irme a la documentacion de android y buscar “como publicar tu app”.

13/04/2021

Hoy vamos a estar intentando eliminar elementos de firebase.

Usaremos una “ventana emergente” para no crear una nueva activity. En la documentacion lo que tenemos que buscar es el “dialogo”, por eso el codigo se llama dialog. El codigo para crear esta ventana es:

```
private void mostrarDialogosSimple(){
    AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());
    builder.setTitle("ELIMINAR RECETA");
    builder.setMessage("Estás seguro de que quieres eliminar esta receta?")
        .setPositiveButton( text: "Eliminar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                //Aqui va todas las opciones para eliminar la receta.
            }
        }).setNegativeButton( text: "Atras", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                //Aqui va las opciones de NO eliminar recetas.
            }
        }).show();
}
```

hay varias formas de dialogos. Para buscar como se hacen los dialogos tengo que mirar en la documentacion a ver como se hacen.

14/04/2021

Mirar como crear un dialogo en la app que para el examen entrará. Mirar alertDialog (cuadros de dialogo). Toolbar, cambiar imagenes. Crear java class, que simplemente es crear la clase, establecerle los atributos, un metodo creador y los metodos get y set:

```

import java.io.Serializable;
public class cRecetas implements Serializable {
    //Atributos de mi clase
    private String idReceta;
    private String cNombre;
    private String cCategory;
    private String cDescripcion;
    private String cFoto;
    private float cValoracion;

    //Método creador o constructor del objeto
    public cRecetas(String id, String nombre, String Cat,
                    String descripcion, String foto, float valoracion) {
        this.idReceta = id;
        this.cNombre = nombre;
        this.cCategory = Cat;
        this.cDescripcion = descripcion;
        this.cFoto = foto;
        this.cValoracion = valoracion;
    }

    //Constructor null para firebase
    public cRecetas(){
    }
}

//Constructor null para firebase
public cRecetas(){
}

```

Y asi se crea la toolbar

```

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

```

La flecha hacia atras de la toolbar se hace asi.

```

//flecha hacia atras
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setDisplayShowHomeEnabled(true);

//método para escuchar los click de la toolbar
toolbar.setNavigationOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //regresar
        finish();
    }
});

```

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    app:title="Recetas"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme"
    app:layout_constraintTop_toTopOf="parent" >

    <TextView
        android:id="@+id/txtToolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Recetas"
        android:textAlignment="center"
        android:textSize="24dp"
        android:textColor="@color/white"

    />

</androidx.appcompat.widget.Toolbar>
```

Para crear un dialogo es simple: creamos un builder y le decimos en que contexto vamos a trabajar, luego le construimos el titulo y luego el mensaje ademas de las opciones (dentro del mensaje salen los dos mensajes).

```
private void mostrarDialogoSimple(){

    AlertDialog.Builder builder = new AlertDialog.Builder( context: detalle.this );
    builder.setTitle("Eliminar recetas");
    builder.setMessage("¿Estás seguro de que quieres eliminar esta receta?")
        .setPositiveButton("Eliminar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
```

Podemos hacer alertDialog estandar pero tambien lo podemos personalizar creando un layout y usarlo como un dialogo con un inflater.

```

private void dialogoPersonalizadoCategorias(){
    AlertDialog.Builder builder = new AlertDialog.Builder( context: altas.this);

    LayoutInflater inflater = getLayoutInflater();
    View view = inflater.inflate(R.layout.activity_altas_categorias, root: null);

    builder.setView(view);

    AlertDialog dialog = builder.create();
    dialog.show();
}

```

Con este metodo puedo crear un solo metodo que me compruebe que boton he tocado y segun lo que haya tocado, haga una cosa. Lo que tengo que hacer es crear una variable b (boton) y asignarle la variable v (que se refiere al View).

```

@Override
public void onClick(View v) {

    Button b = (Button) v;

    Toast.makeText(getApplicationContext(), text: "Pulsado "+b.getText(),Toast.LENGTH_SHORT).show();
}

```

Esto es un dialogo que funciona. OJO CUIDADO CON EL CONTEXTO, NO HAY QUE USAR getApplicationContext! Hay que usar la pantalla en la que estas + .this

```

private void mostrarDialogo(){
    AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
    builder.setTitle("BIENVENIDO");
    builder.setMessage("¿Quieres entrar en la base de datos?").setPositiveButton( text: "SI", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Intent intent = new Intent(getApplicationContext(), pruebabdd.class);
            startActivity(intent);
        }
    }).setNegativeButton( text: "NO", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            //AlertDialog cancelDialog = new AlertDialog.Builder(getApplicationContext()).create();
            dialog.dismiss();
        }
    });
    builder.show();
}

```

16/04/2021

Hoy estaremos con Unity y con open GL.

los motores de carga en tiempo real son Unreal y Unity. Siempre asociamos estos programas a videojuegos pero se usan para muchisimas mas cosas (realidad aumentada, realidad virtual).

Vamos a bajarnos unity. Cuando instalamos Unity no nos descargamos el programa entero, sino que nos descargamos un hub con lo basico y luego tenemos que bajarnos las librerias que vayamos necesitando.

Personal

Comienza a crear con la versión gratuita de Unity

Gratis

Comencemos

Conoce más

Requisitos:

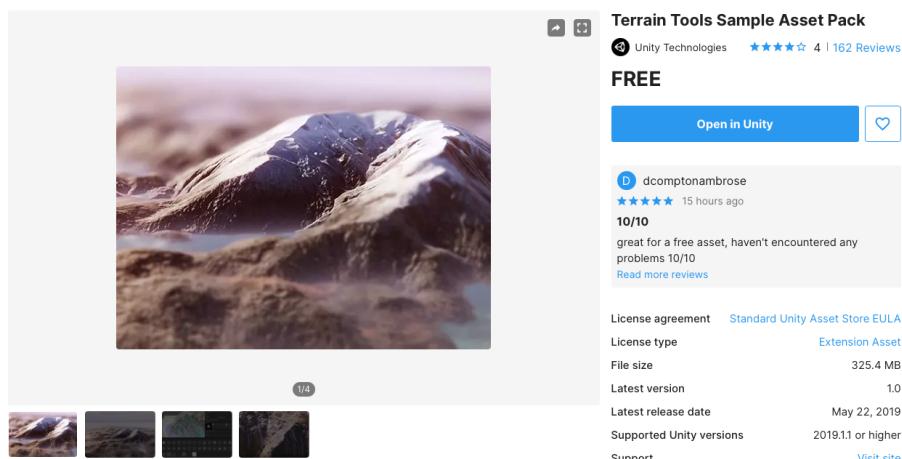
Ingresos o fondos inferiores a USD 100 mil en los últimos 12 meses

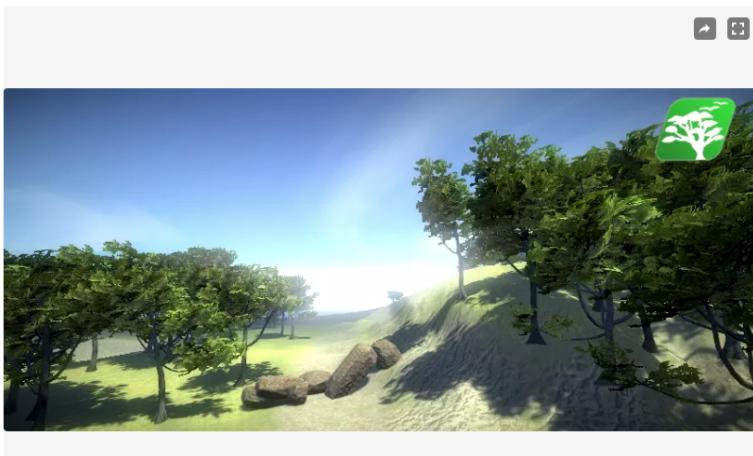
Normalmente los videojuegos lo que hacen es fijar un personaje en el centro de la pantalla y mover el entorno a su alrededor. Tenemos que aprender a usar los gizmo, que es para la camara y el personaje. Lo mas jodido de hacer es el personaje, una vez hecho el personaje ya lo demás es "fácil".

Un sitio donde podemos sacar las esculturas en 3d es una pagina que se llama cults3d y paginas que pueden estar bien para crear personajes, como 3ds max y blender (esta ultima es gratis). Las figuras en 3d estan hechas de triangulos y cuantos mas tenga, mas detalle va a tener y mas problemas va a dar para cargarlo bien. Nosotros vamos a trabajar en lowpoly, que es usar la menor cantidad posible de triangulos.

Unity funciona con scripts programados en C#. Tenemos que conectar Unity, Firebase y Android. Cuando se me instale Unity tengo que configurarlo para que el programa sepa que va a ser ejecutado bajo la supervisión de Android.

Mirar Unity Assets Store (es como una tienda de objetos de Unity). Unos asset chulos para empezar con estos tres:





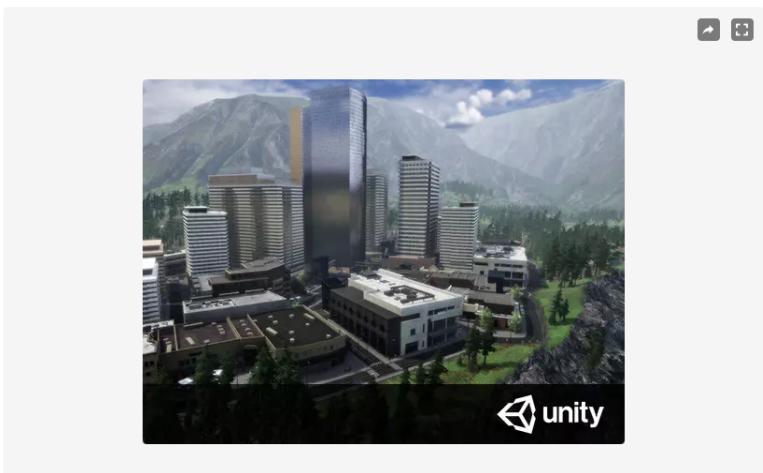
Realistic Tree 9 [Rainbow Tree]

Pixel Games ★★★★★ 4 | 59 Reviews

FREE

Add to My Assets Heart

License agreement	Standard Unity Asset Store EULA
License type	Extension Asset
File size	15.5 MB
Latest version	1.0
Latest release date	Feb 24, 2016
Supported Unity versions	5.3.1 or higher
Support	Visit site



Windridge City

Unity Technologies ★★★★★ 4 | 98 Reviews

FREE

Open in Unity Heart

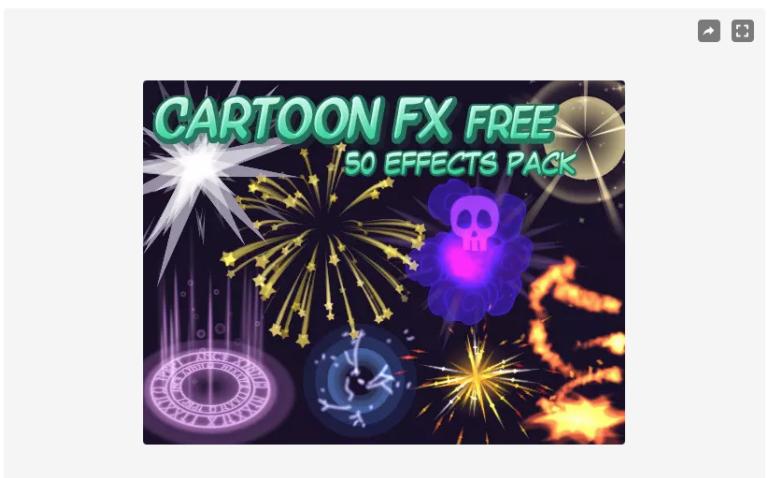
M miguel0583 ★★★★★ 16 days ago

Awesome asset

It is a beautifully built city. I'm eager to try it on VR. I had the white pattern problem in the roads too. I found a quick work around here <https://f...>

[Read more reviews](#)

License agreement	Standard Unity Asset Store EULA
License type	Extension Asset



Cartoon FX Free

Jean Moreno (JMO) ★★★★★ 5 | 257 Reviews

FREE

Add to My Assets Heart

J junneru10 ★★★★★ 2 days ago

Very beautiful vfx pack!

It's amazing that this kind of visual effects are for free! Thank you so much!

[Read more reviews](#)

License agreement	Standard Unity Asset Store EULA
License type	Extension Asset
File size	2.6 MB

19/04/2021

Empezamos con iOS con los lenguajes de Objective C y Swift (nos centraremos en Swift porque es el ultimo y mas actual).

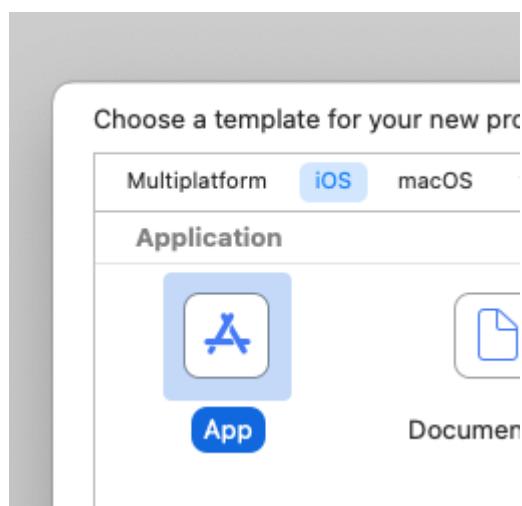
Mientras que en android teníamos una APK, en iOS necesitamos un id de Apple. El id de apple es la cuenta como desarrollador y sin ello no podemos subirla a la web, evidentemente. Para desarrollar apps para apple NECESITAS un Mac, y si tienes windows tienes que bajarte una maquina virtual y es un tema que parece complejo.

Tenemos que instalar XCode y siempre que nos pida ID de apple tenemos que meterle el mail de usuario y la contraseña que tengamos.

Diferencias en android e iOS:

- los layouts en android se llaman vistas en iOS.
- la carpeta res en android se llama assets en iOS.
- el spinner en android se llama picker en iOS.
- hint en android se llama placeholder en iOS.

Cuando creamos un proyecto nuevo tenemos que marcar iOS y luego app para hacer específicamente una app para sistemas operativos iOS.



Apple siempre quiere lo ultimo de lo ultimo, por lo que siempre que tenemos que programar algo tenemos que hacerlo usando la ultima version de todo. Y si la app no funciona en un móvil un poco antiguo, pues unlucky para ti.

En iOS hay 2 pantallas iniciales:

- el launchScreen es una pantalla de carga, después de que se cargue la pantalla se muestra la primera vista de la app.
- la primera vista de la app es el Main.StoryBoard.

Las variables no hay que ponerle el tipo de variable que es, pero en cuanto dos variables distintas interactúen, tienes que decir que clase de variables son. Si digo que num1 es un Int y num2 es un Double y los sumo, da error.

```
//las variables se declaran con var y luego el nombre, no le tengo que decir que tipo de variable (como javascript)
//variables

var num1:Double=8
var num2:Double=3.1

//constantes (se declaraan con let);

let nombre = "Jamon"

var resultado = Double(num2) + Double (num1)
```

Las tuplas y las arrays estan muy relacionadas. Las tuplas son un tipo de dato que sirve para meter y sacar valores de una base de datos. Las tuplas es como si fuera un objeto pero que ocupa mucha menos memoria que un objeto.

Para insertar datos en un array se hace asi.

```
25
26 //añadir nuevos elementos al final
27
28 tablumeros.append(newElement: Int)
29
30 //insertar nuevos elementos en un array en una posicion en concreto
31
32 tablumeros.insert(newElement: Int, at: Int)
33
```

20/04/2021

con las estructuras no creamos objetos porque lo que hacemos es una copia que no tiene instancia. Si yo creo una estructura de un rectangulo y creo un nuevo rectangulo y luego le cambio los atributos a la copia de la estructura, como no tiene instancia lo que hago es cambiarle los atributos a la estructura en general, no a la copia en concreto.

Al final del todo del codigo podemos seguir añadiendo cosas (fuera del codigo) y podemos poner cosas que van a estar escuchando de forma continua.

Los spinner en android se llaman picker y para que un picker funcione necesito varias cosas:

- declararlo tanto a el como al array de donde vaya a sacar los datos

```
@IBOutlet weak var pckMiPicker: UIPickerView!

//variables para todas las acciones
private let misMonedas = ["Euro", "Dolar","Libra","Dinar","Peso"]
```

- declarar el delegate y el datasource:

```
//Picker  
pckMiPicker.dataSource = self  
pckMiPicker.delegate = self
```

- Y abajo, fuera del código, dentro de extensiones, tengo que darle las últimas órdenes.

```
//extensiones  
extension ViewController: UIPickerViewDataSource,  
UIPickerViewDelegate{  
    func numberOfComponents(in pickerView: UIPickerView) -> Int {  
        //Columnas de mi componente (esta vez solo quiero 1)  
        return 1  
    }  
  
    func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {  
        //Filas de mi componente (esta vez solo quiero 1)  
        return misMonedas.count  
    }  
  
    func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String? {  
        //Añadir el título a cada ítem del picker  
        return misMonedas[row]  
    }  
}
```

- y luego ya si queremos meterle que pase algo cuando termine de seleccionar es el método:

```
func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int) {  
    let seleccionado = misMonedas[row]  
    btnNuevo.setTitle(seleccionado, for: .normal)  
}
```

21/04/2021

Para recoger un dato de un texto tengo que usar .delegate.

Si yo quiero pasar una variable de un tipo a otro tipo y el formato:

- var double = Double(String)

Lo que tengo que hacer es añadir exclamaciones al final de la expresion:

- var double = Double(String!)!

Siempre que queramos hacer una copia de un elemento NO USAREMOS EL CONTROL + C NI EL CONTROL + V, lo que tenemos que hacer dejar pulsado option y arrastrar. De esta forma se crea un nuevo elemento que está instanciado y nos va a dar muchos menos problemas que el copiar y pegar.

Para la practica del conversor vamos a usar un segment. Para declarar un segment primero tengo que borrarlo entero, y luego meterle un boton por cada elemento de un array tengo que hacer el array y luego hacer un bucle for con la orden insert.

```
//variables para todas las acciones
private let misMonedas = ["Euro",
    "Dolar","Libra","Dinar","Peso"]
```

```
//Segmento
sgmMiSegmento.removeAllSegments()
for (id,value) in misMonedas.enumerated(){
    sgmMiSegmento
        .insertSegment(withTitle: value,
                      at: id, animated: true)

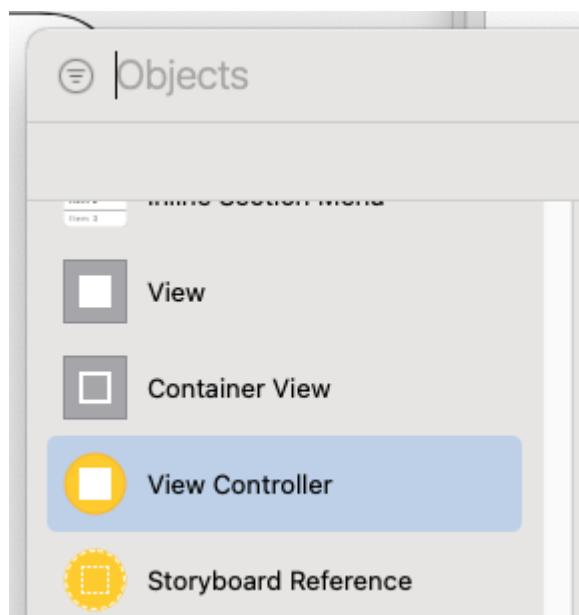
}
```

Y con esto lo que hago es seleccionar el boton que ha sido pulsado:

```
@IBAction func sgmMiSegmentoAction(_ sender:  
Any) {  
  
    pckMiPicker.selectRow(sgmMiSegmento  
        .selectedSegmentIndex, inComponent:  
        0, animated: true)  
}
```

Ahora vamos con navegacion

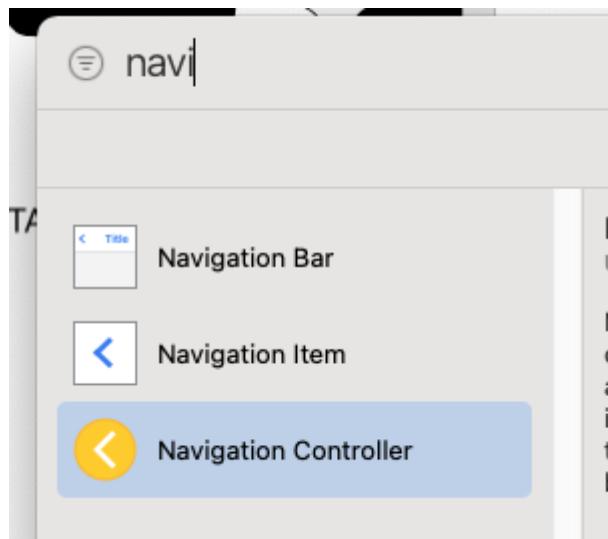
Para crear una pantalla nueva tenemos que darle a nuevo elemento y buscar un view controller.



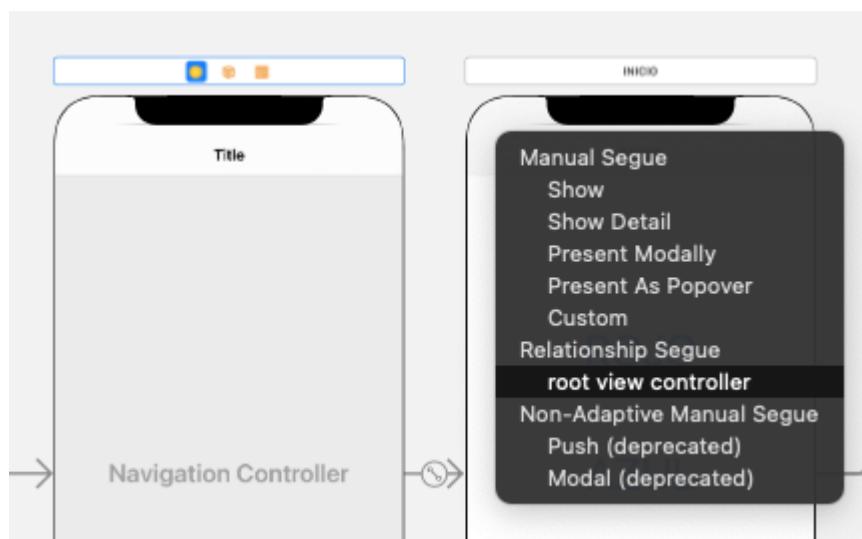
La pantalla nueva va a ser practicamente identica a la original de modo que le cambiaremos el titulo para saber donde cojones estamos.



Las pantallas de iOs funcionan como una pila y se van superponiendo unas encimas de otras. Para poder gestionar bien las pantallas necesito el componente navigation controller.

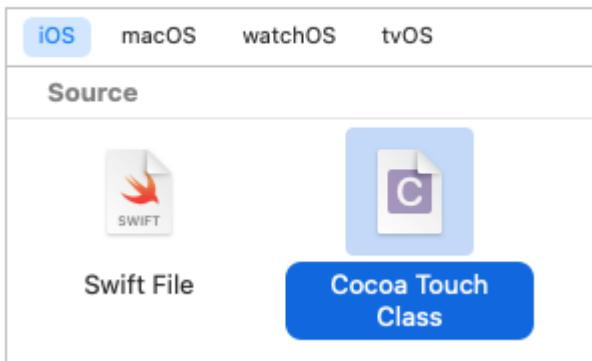


Despues de poner el navigation controller, tengo que ponerlo a él mismo con pantalla principal y luego ponerle como root view controller.

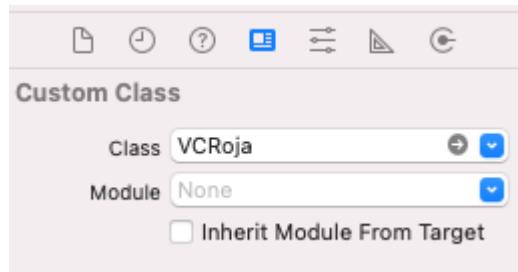


Si quiero crear una clase para que gestione un view controller tengo que crear un Cocoa Touch Class, porque al crearlo voy a poder vincularlo a los view controller. Si abro un archivo Swift File va a estar completamente vacío. El tipo de Cocoa Touch Class tiene que ser UIViewController.

Choose a template for your new file:



Para vincular el Cocoa con una nueva pantalla de iOS tengo que vincular esa pagina al cocoa class poniendole en Custom Class a la pantalla que archivo tiene vinculado.



22/04/2021

Con esto se cambia el tamaño y el grosor de la letra.

```
celda.textLabel?.font = UIFont.systemFont(ofSize: 20, weight: UIFont.Weight.medium)
```

Para enviar un dato tengo que crear un segue y luego se ejecuta con un performSegue.

```
//prepararme el dato para ser enviado
override func prepare(for segue: UIStoryboardSegue,
    sender: Any?) {
    if segue.identifier == "haciaDetalle"{
        if let destino = segue.destination as? VCDetalle {
            destino.mostrar = mostrarItem
        }
    }
}
```

23/04/2021

Para crear objetos es mejor usar el Swift File en lugar del Cocoa Touch, pero es preferible el Swift File porque es un archivo vacío limpio.

esta es la estructura de un objeto en Swift con sus variables y su constructor:

```
7
8 import Foundation
9 import UIKit
10
11 class cArticulo {
12     private var id:Int
13     private var titulo:String
14     private var descripcion:String
15     private var foto:UIImage
16
17     init(id:Int,titulo:String,descripcion:String,foto:UIImage) {
18         self.id = id
19         self.titulo = titulo
20         self.descripcion = descripcion
21         self.foto = foto
22     }
23
24     func setId(id:Int){self.id = id}
25     func setTitulo(titulo:String){self.titulo = titulo}
26     func setDescripcion(descripcion:String){self.descripcion = descripcion}
27     func setFoto(foto:UIImage){self.foto = foto}
28
29     func getId() -> Int {return self.id}
30     func getTitulo() -> String {return self.titulo}
31     func getDescripcion() -> String {return self.descripcion}
32     func getFoto() -> UIImage {return self.foto}
33 |
34 }
35
```

26/04/2021

Cuando creamos una celda personalizada no deberemos tocar la anchura de la celda porque a lo mejor lo toco y se me sale de la pantalla del teléfono. El alto puedo tocarlo sin problema pero el ancho no.

Además, todos los elementos los tengo que declarar en el archivo asociado a la celda, no en el archivo asociado a la tabla. Es decir, para usar una celda personalizada tengo que declarar sus elementos dentro del archivo que gestiona la celda, y luego en la pantalla asociada al tableView tengo que registrar la celda para poder usarla.

```
//Aqui le digo que puedo usar la celda personalizada
//unib es la vista asociada
//Bundle lo pongo en nil porque estoy dentro del
//propio
//proyecto que llama a esa celda
tableView.register(UINib(nibName:
    "celdaPersonalizada", bundle: nil),
    forCellReuseIdentifier: "miCeldaPersonalizada")
```

Recordatorio:

Para crear bien una conexión entre pantallas tengo que crear un camino entre las dos pantallas desde el viewControler. También tengo que crear un archivo controlador para la nueva pantalla.

MVC => Modelo Vista Controlador.

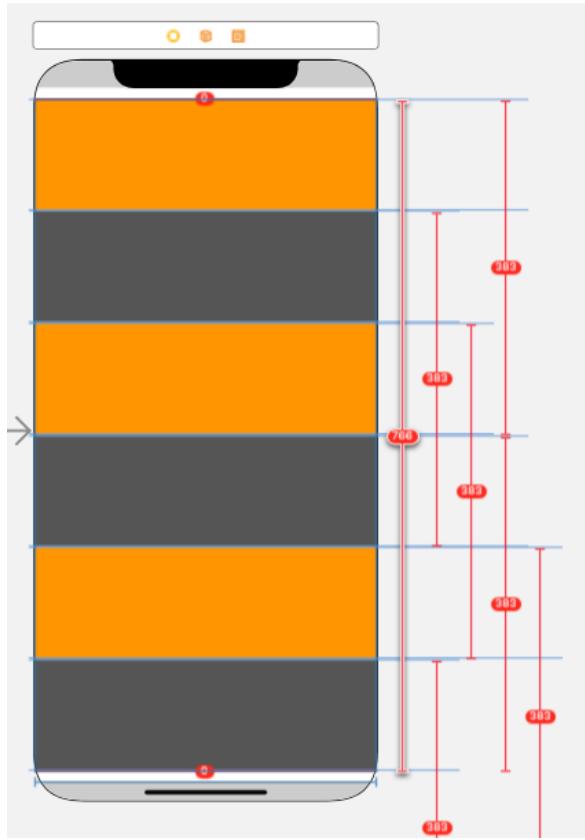
27/04/2021

Cuando ponemos un scroll view tenemos que quitarle el tick en content layout guides en ajustes del elemento para que no se me ajuste a ninguna guía de nada. Después, tenemos que añadir los constraint a las 4 partes a 0.

Los pasos para poner un scroll view es:

- poner el scroll view y ponerle las contrains a 0
- ponerle una view encima del scroll view y ponerla a 0 de constrain, ajustar al máximo de la pantalla y luego ponerle la constrain de width y height (se conoce como false view)
- ponerle el stack view encima de la view anterior y ponerle a 0 todos los constrains
- después de eso, ya puedo meter las view que quiera para que hagan el scroll pero es importante señalar que tengo que ponerle las constrain interiores, las que marcan el width y height.

IMPORTANTE: cuando después de meterle vistas en el scroll view me va a dar problemas porque la primera vista que le metí se ha quedado con la constrain ocupando la pantalla entera y tengo que eliminar la constrain grande (la que está marcada).



Una vez que la quite, ya se queda ajustada bien.

Si lo que quiero es ver paginas web en el movil tengo que ponerle una vista que se llama webkit view. Lo mismo, lo pongo en el view controller y le pongo las constrain a 0. Despues de crear el view controller y de ponerle el webkit view, le creo una clase con el cocatouch.

los webkit tienen su propia libreria y hay que importarla antes de usarla, porque si no dara error.

Para añadir maps hay que ponerle el componente map kit view e importar la libreria en el view controller. Para centrarlo en un punto en concreto tengo que llamar a esta clase:

```
// aqui doy unas coordenadas
miMapa.centerCoordinate = CLLocationCoordinate2D(latitude: 27.175192505649047,
longitude: 78.04213146879961)

//Añadir un punto de localizacion al mapa
let localizacion = MKPointAnnotation()

localizacion.coordinate = CLLocationCoordinate2D(latitude: 27.175192505649047,
longitude: 78.04213146879961)

localizacion.title = "Aqui esta el Taj Majal"

//Añadir localizacion al mapa
miMapa.addAnnotation(localizacion)
```

```

//Crear region con argumentos específicos (para que el mapa me aparezca mas cerca)
let region = MKCoordinateRegion(center: localizacion.coordinate,
    latitudinalMeters: 100, longitudinalMeters: 100)

//Añadir region al mapa
miMapa.setRegion(region, animated: true)

miMapa.mapType = .satellite

```

Para hacer botones de zoom in y zoom out hay que hacer 2 variables de latitud y longitud y una nueva variable de region

```

@IBAction func btnPlusAction(_ sender: Any) {

    if latitudmetros > 0 && longitudmetros > 0 {
        latitudmetros-=50
        longitudmetros-=50
    }

    let nueva = MKCoordinateRegion(center: localizacion.coordinate, latitudinalMeters:
        CLLocationDistance(exactly: latitudmetros)!, longitudinalMeters: CLLocationDistance(exactly:
        longitudmetros)!)

    //Añadir region al mapa
    miMapa.setRegion(nueva, animated: true)

}

@IBAction func btnMinusAction(_ sender: Any) {
    if latitudmetros < 5000 && longitudmetros < 5000 {
        latitudmetros+=50
        longitudmetros+=50
    }

    let nueva = MKCoordinateRegion(center: localizacion.coordinate, latitudinalMeters:
        CLLocationDistance(exactly: latitudmetros)!, longitudinalMeters: CLLocationDistance(exactly:
        longitudmetros)!)

    //Añadir region al mapa
    miMapa.setRegion(nueva, animated: true)
}

```

28/04/2021

Cuando declaro cosas tengo que preguntarme si voy a mandarle hacer cosas o modificar sus atributos. Si quiero modificar sus atributos tengo que declararlo arriba del código, en los outlets, y si quiero que haga cosas tengo que declararlo debajo del onCreate (y lo declaro en los dos sitios si quiero hacer las dos cosas).

para recoger un dato de los editText que pongo en una pantalla tengo que asignarle un delegate, que es un protocolo, no una función.

Siempre que añada una extensión, tengo que ponerle extensión y luego el nombre del propio controlador que estoy usando.

```
//Recoger dato de los textos
etxtTitulo.delegate = self
etxtDescripcion.delegate = self
```

Esta es la estructura general de los delegate del textView y del textField

```
51
52 extension VCAltas: UITextFieldDelegate, UITextViewDelegate{
53     //recogo los textos
54     func textFieldShouldReturn(_ textField: UITextField) -> Bool {
55         //Aqui muestro el teclado
56         textField.resignFirstResponder()
57     }
58
59     func textFieldDidEndEditing(_ textField: UITextField) {
60         //Aqui es donde guardo el dato para el alta
61     }
62
63     func textViewShouldEndEditing(_ textView: UITextView) -> Bool {
64         textView.resignFirstResponder()
65     }
66
67     func textViewDidEndEditing(_ textView: UITextView) {
68         //Guardar campo de descripcion
69     }
70
71     func textViewDidBeginEditing(_ textView: UITextView) {
72         etxtDescripcion.text = ""
73         etxtDescripcion.textColor = .black
74     }
75
76 }
77
78
79 }
80
```

Para acceder a la camara del telefono tenemos que coger un pickerController para que seleccione la imagen.

Yo en pantalla no tengo ningun picker y lo añado de forma dinamica con una constante:

```
//Imagen
let imagePicker: UIImagePickerController = UIImagePickerController()
```

hago un delegate para coger la imagen de ese picker:

```
//Recoger la foto
imagePicker.delegate = self
```

Las extensiones relacionadas con el delegate no las metemos como extensiones porque pueden dar problemas, asi que las ponemos al iniciar la clase viewController:

```
9  
10 class VCAltas: UIViewController, UIImagePickerControllerDelegate & UINavigationControllerDelegate {  
11
```

Con esto accedemos a la camara

```
//Acceder a la camara  
@IBAction func btnHacerFotoAction(_ sender: Any) {  
  
    //Comprobaciones para acceder a la camara  
    //ver si la camara esta disponible  
    if(UIImagePickerController.isSourceTypeAvailable(.camera)){  
        //comprobar si la camara de atras esta disponible  
        if (UIImagePickerController.availableCaptureModes(for: .rear) != nil) {  
  
            //No permitir editar imagen  
            imagePicker.allowsEditing = false  
            //indicar si la foto se recoge de la camara o de la galeria (esto lo tiene que saber la maquina)  
            imagePicker.sourceType = .camera  
            //que tipo de archivo es  
            imagePicker.cameraCaptureMode = .photo  
  
            //Esta accion es la que abre la camara y coge la foto  
            present(imagePicker, animated: true, completion: nil)  
        }  
    }  
}
```

Y con esto accedo a la galeria

```
//Acceso a la galeria  
@IBAction func btnAddAction(_ sender: Any) {  
  
    //Comprobaciones para acceder a la camara  
    //ver si la galeria esta disponible  
    if(UIImagePickerController.isSourceTypeAvailable(.photoLibrary)){  
  
        //No permitir editar imagen  
        imagePicker.allowsEditing = false  
        //indicar si la foto se recoge de la camara o de la galeria (esto lo tiene que saber la maquina)  
        imagePicker.sourceType = .photoLibrary  
  
        //Esta accion es la que abre la camara y coge la foto  
        present(imagePicker, animated: true, completion: nil)  
    }  
}
```

Y con esto guardo la imagen

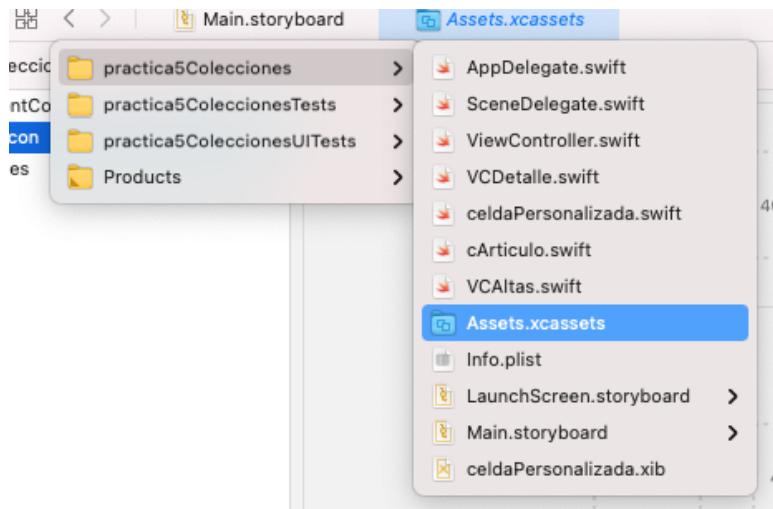
```
//Esta funcion lo que hace es recoger la foto y guardarla en la galeria.
func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {
    //guardo en la constante la imagen original seleccionada
    if let imagenSeleccionada: UIImage = info[UIImagePickerController.InfoKey.originalImage] as? UIImage{
        articuloAlta.setFoto(foto: imagenSeleccionada)

        //comprobar si el archivo viene de la galeria o de la camara
        if imagePicker.sourceType == .camera {
            //guardar la foto de la camara en la galeria
            UIImageWriteToSavedPhotosAlbum(imagenSeleccionada, nil, nil, nil)
        }

        imgFoto.image = imagenSeleccionada
        articuloAlta.setFoto(foto: imagenSeleccionada)
        //cerrar el picker dinamico y liberar memoria
        imagePicker.dismiss(animated: true, completion: nil)
    }
}
```

29/04/2021

Para ponerle un icono a la app hay que hacerlo aqui:



Para poner la imagen bien bien lo mejor es sacarla a 1024 y que la app la gestione como toque. Para sacar imagenes bien para las apps de iOs voy a sacar la imagen de una web que se llama appicon.co (es inglesa o de por ahí). En esta pagina subo la imagen y le doy a descargar, me descarga un zip y para los iconos de iOs tengo que usar la carpeta de assets.xcassets.

Para hacer un segue de vuelta de una pantalla, tengo que hacer en la pantalla de origen una funcion que se llama unwind, que es basicamente para usar un segue a la inversa. Por ejemplo, en el caso de la app de colección tengo que enviar un dato de la pantalla 1 a la 2 y luego de la 2 a la 1 tengo que enviar otro objeto, por lo que en la pagina 1 tengo que añadir la funcion de vuelta.

```

//Funcion para usar el segue de regreso
@IBAction func unwindToAltas(_ unwindSegue: UIStoryboardSegue) {
    _ = unwindSegue.source
    // Use data from the view controller which initiated the unwind segue
    //Aqui pongo el objeto que mando para que la pantalla anterior lo recoja
    if (recogerArticulo != nil){

        //Guardar el id del objeto que traigo para que sea el ultimo elemento que introduzco
        recogerArticulo?.setId(id: miColeccionObj.count + 1)
        //Aqui he añadido un nuevo articulo a mi array
        miColeccionObj.append(recogerArticulo!)
        //Tengo que actualizar el tableView con los datos que acabo de meterle
        tableView.reloadData()

    }
}

```

En la pagina que envia el objeto tambien tengo que hacer un prepare para enviar el objeto.

```

//Esta funcion prepara los datos enviados por un segue que se ejecuta
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    let destino = segue.destination as? ViewController
    destino?.recogerArticulo = articuloALta
}

```

Una vez que tenga el codigo configurado, ya tengo la opcion de arrastrar el boton de regreso hacia la casilla exit del controlador de la segunda pantalla, de la pantalla que usa el segue para volver.



En android tenemos el manifest pero en iOs tenemos el infoplist. Ahí tenemos que subir los permisos que queramos pedir al usuario y tenemos que añadir tambien una descripcion de para que queremos ese permiso.

Information Property List	Dictionary	(19 items)
Privacy - Media Library Usage Description	String	"voy a seleccionar fotos de la librería para utilizarlo en mis artículos"
Privacy - Photo Library Additions Usage Description	String	"para que se guarden las fotos"
Privacy - Camera Usage Description	String	"utilizo la cámara para guardar las fotos de mis artículos"

Para que una imagen se estire y quede centrada le tengo que cambiar este sitio a Aspect Fill



Esto es el codigo de un alert con dos opciones, eliminar o cancelar. Un alert NO ES UN TOAST, es una ventana de dialogo de android.

```
//Esta funcion la he hecho yo, no esta en el sistema
func mostrarAlert(titulo: String, mensaje: String, b1: String, b2: String){

    //Aqui defino el titulo y el mensaje de mi alert o ventana pop up
    let alert = UIAlertController(title: titulo, message: mensaje, preferredStyle: .alert)

    //Aqui pongo los botones
    let accion1 = UIAlertAction(title: b1, style: .cancel)
    let accion2 = UIAlertAction(title: b2, style: .default)

    //añadir acciones a mi alert
    alert.addAction(accion1)
    alert.addAction(accion2)

    //Esto muestra el alert al usuario
    present(alert, animated: true, completion: nil)

}
```

30/04/2021

Empezamos hablando del infolist de iOs, ese archivo que es el equivalente al manifest de Android. En cada permiso que pida en el infolist tengo que poner en un string la explicacion (preferiblemente en ingles) de por que pido ese permiso.

Ahora vemos los tipos anidados y los protocolos.

las estructuras anidadas lo que me permiten es crear objetos dentro de unas directrices. Por ejemplo:

```

struct piezaAjedrez {
    let color: Color
    let tipo: Tipo

    enum Color:String {
        case blanca = "Blanca"
        case negra = "Negra"
    }

    enum Tipo:String {
        case rey = "Rey"
        case reina = "Reina"
        case alfil = "Alfil"
        case caballo = "Caballo"
        case torre = "Torre"
        case peon = "Peon"
    }

    struct numPiezas {
        let numPiezas: Int
    }
    var numpiezas: numPiezas{
        switch self {
            case .rey:
                return numPiezas(numPiezas: 1)
            case .reina:
                return numPiezas(numPiezas: 1)
            case .alfil:
                return numPiezas(numPiezas: 2)
            case .caballo:
                return numPiezas(numPiezas: 2)
            case .torre:
                return numPiezas(numPiezas: 2)
            case .peon:
                return numPiezas(numPiezas: 8)
        }
    }
}

```

Esta es la estructura de una pieza de ajedrez, y lo que hace es que yo, al crear el objeto piezaAjedrez NO ME DEJE CREARLA COMO A MI ME DE LA GANA, sino que tiene que ser dentro de unas directrices y de ahí no se puede salir.

los protocolos lo que hacen es darme un mínimo de requisitos a la hora de crear estructuras. Es decir:

```

protocol personasProtocol{
    var nombre: String { get set }
    var edad: Int { get set }

    func nombreCompletar() -> String
}

struct alumno: personasProtocol {
    var nombre: String
    var edad: Int
    var curso: String

    func nombreCompletar() -> String {
        return nombre
    }
}

```

Aquí tengo el protocolo personas y le digo que, MINIMO, me tiene que dar un nombre, una edad y la función completar nombre. Al crear la estructura alumno y extenderla de personasProtocol, lo que hago es crear una persona que va a ser como a mí me de la gana PERO tiene que tener MINIMO los parámetros que le he dado en el protocolo. Por ejemplo, usando el protocolo personasProtocol puedo crear 2 estructuras, alumno y profesor.

```

struct alumno: personasProtocol {
    var nombre: String
    var edad: Int
    var curso: String

    func nombreCompletar() -> String {
        return nombre
    }
}

struct profesor: personasProtocol {
    var nombre: String
    var edad: Int
    var cursoqueimparte: String
    var temario: String

    func nombreCompletar() -> String {
        return nombre
    }
}

var Alumno = alumno(nombre: "Javier", edad: 30, curso: "iOs")
var Profesor = profesor(nombre: "Juan", edad: 37, cursoqueimparte: "Apps móviles", temario: "iOs")

```

Para eliminar una celda de la tableview de la app de ropa se hace asi.

```

//Eliminar celda en el tableview. Aquí saco el estilo que va a tener el botón de
borrado
func tableView(_ tableView: UITableView, editingStyleForRowAt indexPath:
    IndexPath) -> UITableViewCell.EditingStyle {
    return .delete
}

//eliminar la celda seleccionada
func tableView(_ tableView: UITableView, commit editingStyle:
    UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        tableView.beginUpdates()
        miColecciónObj.remove(at: indexPath.row)
        tableView.deleteRows(at:[indexPath], with: .fade)
        tableView.endUpdates()
    }
}

```

Y con esto modiflico el titulo del boton eliminar

```

//para cambiar el título del botón delete
func tableView(_ tableView: UITableView, titleForDeleteConfirmationButtonForRowAt
    indexPath: IndexPath) -> String? {
    return "Vete de aquí."
}

```

Ahora, para hacer que el tableview pueda editarle las posiciones se hace asi.

```

//con esto me permite mover las celdas
func tableView(_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {
    return true
}

//Cambiamos el orden de la celda
func tableView(_ tableView: UITableView, moveRowAt sourceIndexPath: IndexPath, to destinationIndexPath: IndexPath) {
    miColeccionObj.swapAt(sourceIndexPath.row, destinationIndexPath.row)
}

//Esto es para ordenar la tabla
@IBAction func OrdenarAction(){
    if tableView.isEditing == true {
        tableView.isEditing = false
    } else {
        tableView.isEditing = true
    }
}

```

IMPORTANTE: la funcion OrdenarAction() hay que ponerla en un bar button (o en donde sea).

Si quiero meter un buscador tengo que meterle el elemento search bar, y ya esta (la parte grafica, tengo que meterle codigo).

La barra de busqueda para poder utilizarla tengo que declararla por cojones y despues, tengo que hacerle un barraBusqueda.delegate = self para que empiece a coger los datos, y ahora dara el error de cuando no implementas el delegado. Al final, lo mejor es poner el delegado y añadirle sus cosas a una extension:

```

extension ViewController: UISearchBarDelegate{
    //Recoger el dato que escribimos en la barra
    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        //al cambiar el texto segun voy escribiendo
        print(searchText)
    }
}

```

Con esto lo que hago es imprimir por la consola el texto que voy escribiendo.

4/05/2021

Hoy vemos como acceder y guardar a los datos guardados en el telefono con sets y conjuntos.

```

//SETS y conjuntos

var miSet = Set<String>()

//inserto valores en conjuntos

//Insertar valores de 1 en 1
miSet.insert("Javier")
miSet.insert("Morillas")
miSet.insert("30")

print(miSet)

//insertar valores todos la vez
var miSet2: Set<String> = ["Javier", "Morillas", "30"]

print(miSet2)

//acceso a un set
if miSet.contains("Javier"){
    print("Existe.")
} else {
    print("No existe.")
}

```

Operaciones con conjuntos

```

//Operaciones de conjuntos
//hago dos conjuntos para que hagan operaciones entre ellos
let miIntSet: Set<Int> = [1,2,3,4,5]
let miIntSet2: Set<Int> = [0,2,4,8,16]

//Interseccion: se queda con los valores que coincidan con ambos conjuntos
print(miIntSet.intersection(miIntSet2))

//Diferencia simetrica: se queda con los valores que no coinciden
print(miIntSet.symmetricDifference(miIntSet2))

//Union
print(miIntSet.union(miIntSet2))

//sustraccion
print(miIntSet.subtracting(miIntSet2))

```

Android usa un sistema de base de datos que se llama SQL y SQLite pero en iOS las cosas se almacenan usando el sistema CoreData.

Las colecciones se ordenan así

```
//algoritmos para colecciones

var miArray = [4,3,5,1,2]

var miDiccionario = [2:"Dos", 5:"Cinco", 3:"Tres", 1:"Uno", 4:"Cuatro"]

var miSet: Set = [4,3,5,1,2]

//esto es un array, un diccionario y un set

//ordenar datos
//orden array de menor a mayor
miArray.sort()

//orden array de mayor a menor
miArray.sort{
    (intA, intB) -> Bool in
        return intA > intB
}
```

```
//los diccionarios no se pueden ordenar con sort()
/*La única forma de ordenar un diccionario es pasarlo a un array,
y luego ordenar el array*/

var miArray2 = [4,3,5,1,2]

var miDiccionario2 = [2:"Dos", 5:"Cinco", 3:"Tres", 1:"Uno", 4:"Cuatro"]

var miSet2: Set = [4,3,5,1,2]

let miDiccionarioOrdenado = miDiccionario2.sorted {
    (intA, intB) -> Bool in
        return intA.key < intB.key
}
```

```
//Ordenar un Set es identico al array  
//de menor a mayor  
var miSetOrdenado = miSet2.sorted()  
  
//de mayor a menor  
miSetOrdenado = miSet2.sorted {  
    (intA, intB) -> Bool in  
        return intA > intB  
}
```

Ahora vamos a hacer un mapa de algoritmos.

```
//sumar 10 a todos los valores de mi array sin tener que recorrerlo.  
//lo hago con un mapa  
let miMapArray = miArray.map{  
    (miInt) -> Int in  
        return miInt + 10  
}
```

Esto lo que hace es crear un segundo array con los datos del primer array y añadirle 10 en mi caso.

```
//mostrar un dato de forma rapida de un array  
let miStringMapaArray = miArray.map{  
    (miInt) -> String in  
        return "Este numero es \(miInt)"  
}
```

Recorrer colecciones

```
//Como recorrer colecciones
//for each en array
miArray.forEach {
    (miInt) in
    print(miArray)
}

//for each en diccionarios
miDiccionario.forEach {
    (miElemento) in
    print(miElemento.key)
}

//for each en sets
miSet.forEach {
    (miElemento) in
    print(miElemento)
}
```

Respecto al tema de la persistencia de datos: necesito guardar datos para que cuando cambie de pantallas no se me borren los datos, y eso lo consigo con el core data y el user defaults.

Para empezar a usar un user default necesito una clave,

Para hacer un alert sencillo que solo diga que ok se hace asi.

```
private func mostrarAlert(mensaje: String){
    let alert = UIAlertController.init(title: "mi almacen de datos", message: mensaje, preferredStyle:
        .alert)
    alert.addAction(UIAlertAction(title: "OK", style: .default, handler: nil))

    present(alert, animated: true, completion: nil)
}
```

Aqui esta el codigo de user default para get, put y delete

```

//Con el get lo que vamos a hacer es recuperar las preferencias del sistema
@IBAction func btnGetAction(_ sender: Any) {

    //Compruebo que lo que estoy buscando existe
    if let valorString = UserDefaults.standard.string(forKey: miKey){
        //mostrar el valor en un alert
        mostrarAlert(mensaje:valorString)
    } else {
        mostrarAlert(mensaje: "No se ha recuperado ningun dato de UserDefaults")
    }

}

//Aqui guardaremos las preferencias del sistema
@IBAction func btnPutAction(_ sender: Any) {

    //Guardar nombre y apellido del usuario
    //acceso
    UserDefaults.standard.set("Javier Morillas", forKey: miKey)

    //Apple guarda los datos cuando quiere, asi que tengo que forzar el guardado
    UserDefaults.standard.synchronize()

}

//aqui eliminaremos las preferencias del sistema
@IBAction func btnDeleteAction(_ sender: Any) {

    //Borrar el string guardado
    UserDefaults.standard.removeObject(forKey: miKey)

    //fuerzo la sincronizacion
    UserDefaults.standard.synchronize()
}

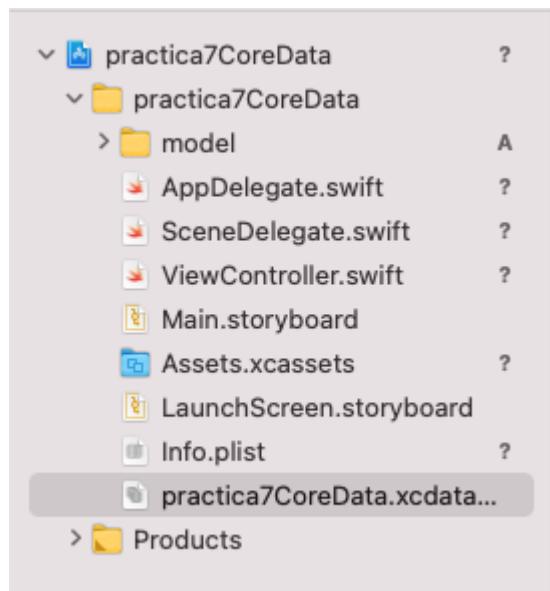
```

05/05/2021

El user default hay que imaginarlo como una tabla en el que pondremos lo que queramos, pero en realidad es como un objeto que se llama Settings en el que vamos a guardar lo minimo para que la app pueda acceder a los datos guardados y que la app pueda funcionar.

El core data es como una base de datos SQLite, pero NO ES LO MISMO. El core data es un framework oficial de apple basado en object graph persistence framework.

Puedo añadir el core data en el momento de crear el proyecto o puedo añadirlo despues. Si lo creo al principio del proyecto me tiene que aparecer este archivo.



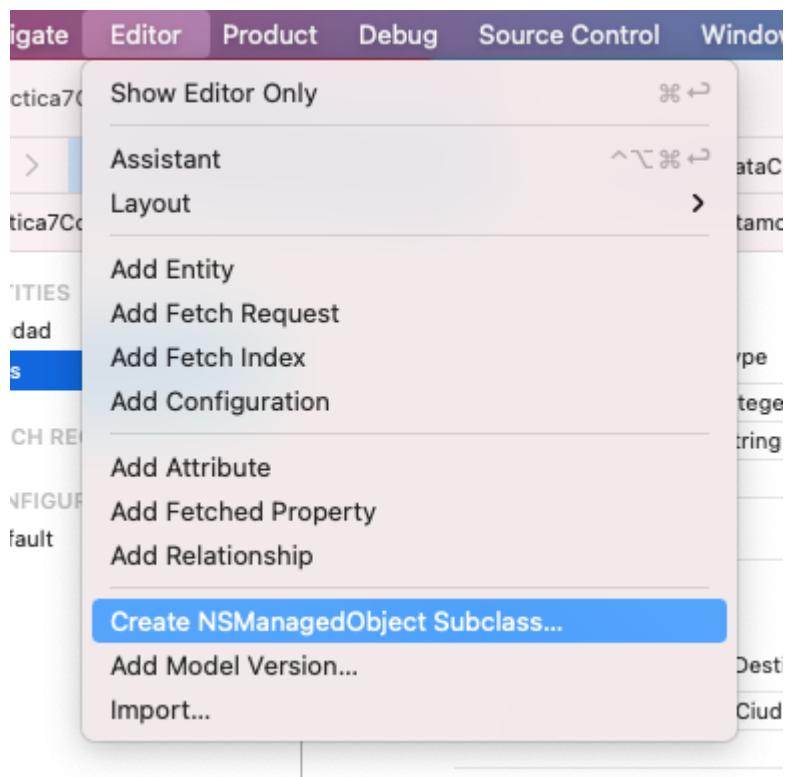
En ese archivo puedo crear entidades que serán como objetos para trabajar con ellos, pudiendo darles atributos y relaciones.

Attribute	Type
[N] idpais	Integer 32
[S] nombrepais	String

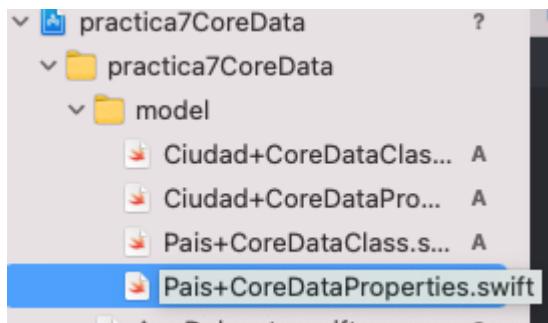
Relationship	Destination	Inverse
[O] paisesciudades	Ciudad	♦ No Inverse ♦

Para crear las clases de las entidades lo que tengo que hacer es darle al panel de la derecha y donde pone codegen lo suyo es ponerle Manual para poder trabajar con el

Luego arriba le doy a editor y a Create NSManagedObject, y me crea una serie de archivos (que luego lo suyo es crear una carpeta y meterlos ahí para tener el código organizado).



Si por lo que sea tengo que modificar las entidades (por añadirles nuevos atributos o lo que sea) tengo que eleminar el archivo class properties y generar el archivo de nuevo.



Si quiero llenar un tableview con la lista de cosas que tengo dentro de un array tengo que declarar el tableview en los outlets y añadirle el dataSource = self. Este metodo es el que carga las cosas. tambien tengo que ponerle el manage de la base de datos

```

@IBOutlet weak var tbMisPaises: UITableView!

//1º referencia managed
/*Esto hace referencia al manager para que se comunique con la base de datos
 y poder usar sus datos. Lo que hago es delegando toda mi app para que pueda usar
 el contenedor persistente*/
let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext

var miPais1 = Pais()
var miPais2 = Pais()
var miPais3 = Pais()
var miPais4 = Pais()
var miPais5 = Pais()

var miArrayPaises: [Pais]?

```

tbMisPaises.dataSource = self

Luego, lo que tengo que hacer es añadir una extensión y ahí pondre el numero de filas que quiero que tenga y le digo que tenga tantas filas como entradas tenga mi array.

```

extension ViewController: UITableViewDataSource{
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return miArrayPaises!.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        var cell = tableView.dequeueReusableCell(withIdentifier: "miCell")

        if (cell != nil){
            cell = UITableViewCell(style: .default, reuseIdentifier: "miCell")
        }
    }
}

```

con estos dos codigos en el tableview se que celda se ha pulsado

tbMisPaises.delegate = self

```

extension ViewController: UITableViewDelegate {

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        print(miArrayPaises![indexPath.row].nombrepais!)
    }
}

```

Con esto elimino la celda seleccionada y su valor de la base de datos

```

//evento para eliminar la celda seleccionada

func tableView(_ tableView: UITableView, trailingSwipeActionsConfigurationForRowAt indexPath: IndexPath) -> UISwipeActionsConfiguration? {
    //crear accion de eliminar
    let eliminar = UIContextualAction(style: .destructive, title: "Eliminar"){
        (action, view, completionHandler) in

        //aqui averiguo qué elemento tengo que eliminar (el que se ha pulsado)
        let paisEliminar = self.miArrayPaises![indexPath.row]

        //Eliminar de mi base de datos
        self.context.delete(paisEliminar)

        //guardar el cambio en la base de datos
        try! self.context.save()

        //recargar los datos
        self.recuperarDatos()
    }

    //ejecuto la accion configurada
    return UISwipeActionsConfiguration(actions: [eliminar])
}

```

Con esto cojo la celda elegida y la modifco.

```

//Con esto selecciono una celda y modifco su contenido
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

    let paisEditar = self.miArrayPaises![indexPath.row]

    let alert = UIAlertController(title: "Modificar", message: "Modificar el nombre", preferredStyle: .alert)
    alert.addTextField()

    let textField = alert.textFields![0]
    textField.text = paisEditar.nombrepais

    let boton = UIAlertAction(title: "Modificar", style: .default){ (action) in
        let textField = alert.textFields![0]
        paisEditar.nombrepais = textField.text

        try! self.context.save()
        self.recuperarDatos()
    }

    alert.addAction(boton)
    self.present(alert, animated: true, completion: nil)
}

```

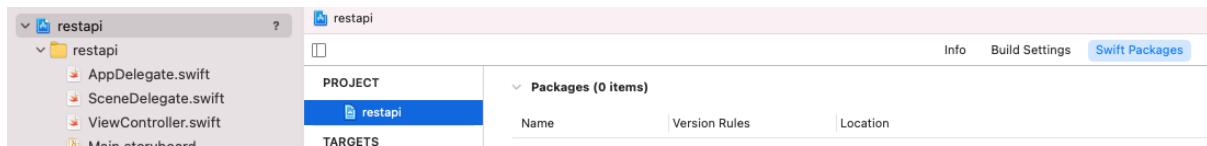
06/05/2021

con la API Rest tengo que conocer dos líneas de código: Code y Meta.

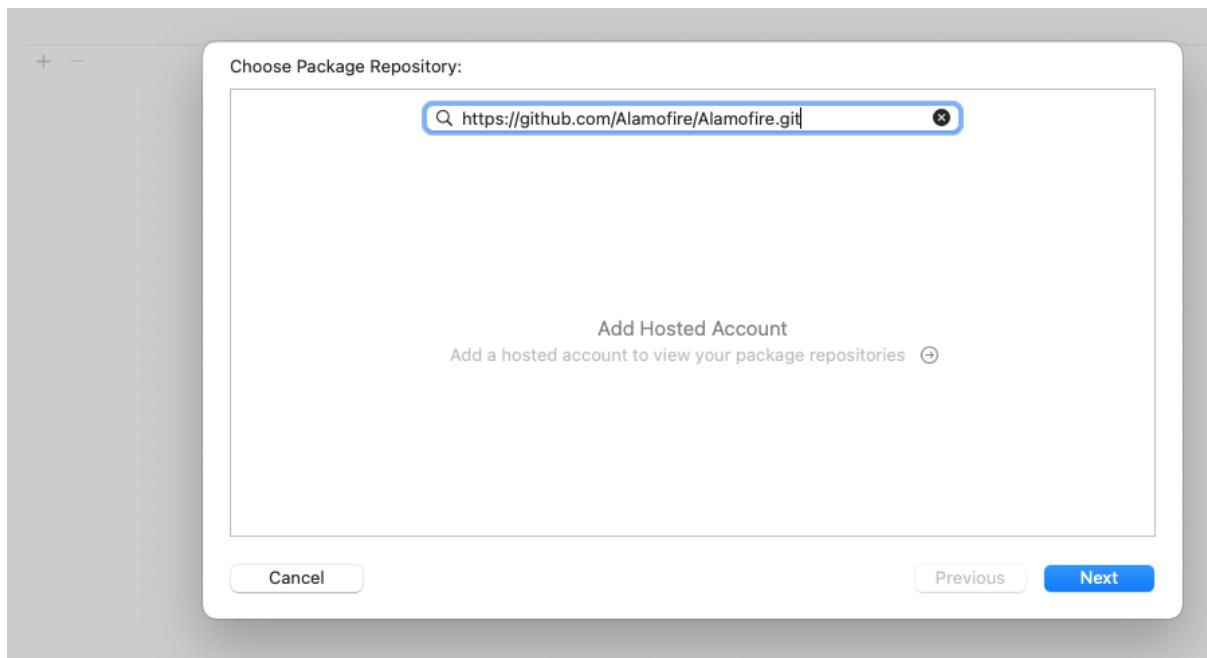
```
        "code": 200,
        "meta": {
            "pagination": {
                "total": 1601,
                "pages": 81,
                "page": 1,
                "limit": 20
            }
        },
    },
```

Aquí me está diciendo que Code es 200, lo que significa que es una respuesta positiva (los códigos de 200 a 299 son positivos). La parte Meta me está diciendo que hay 1601 registros divididos en 81 páginas, estoy en la página 1 y me está diciendo que cada página muestra 20 registros.

Las librerías en Swift se instalan así:



Luego le doy al '+' y añado el link de la librería de GitHub:



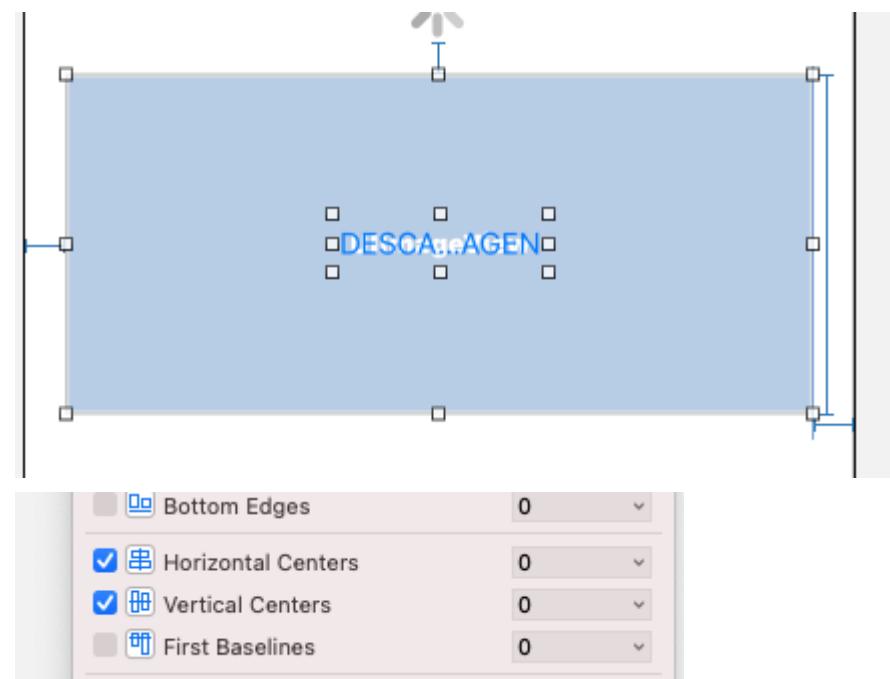
después de instalar la librería tengo que hacer un archivo de Swift que se va a llamar provider:

IMPORTANTE: en un JSON solo puedo guardar tipos primitivos, si quiero meter datos que no sean primitivos (como una fecha), tengo que hacer una pirula.

10/05/2021

seguimos con restapi

Para meter un elemento dentro de otro tengo que meterlo, seleccionar los dos elementos y al poner las constraints se tomara el elemento padre como guia.

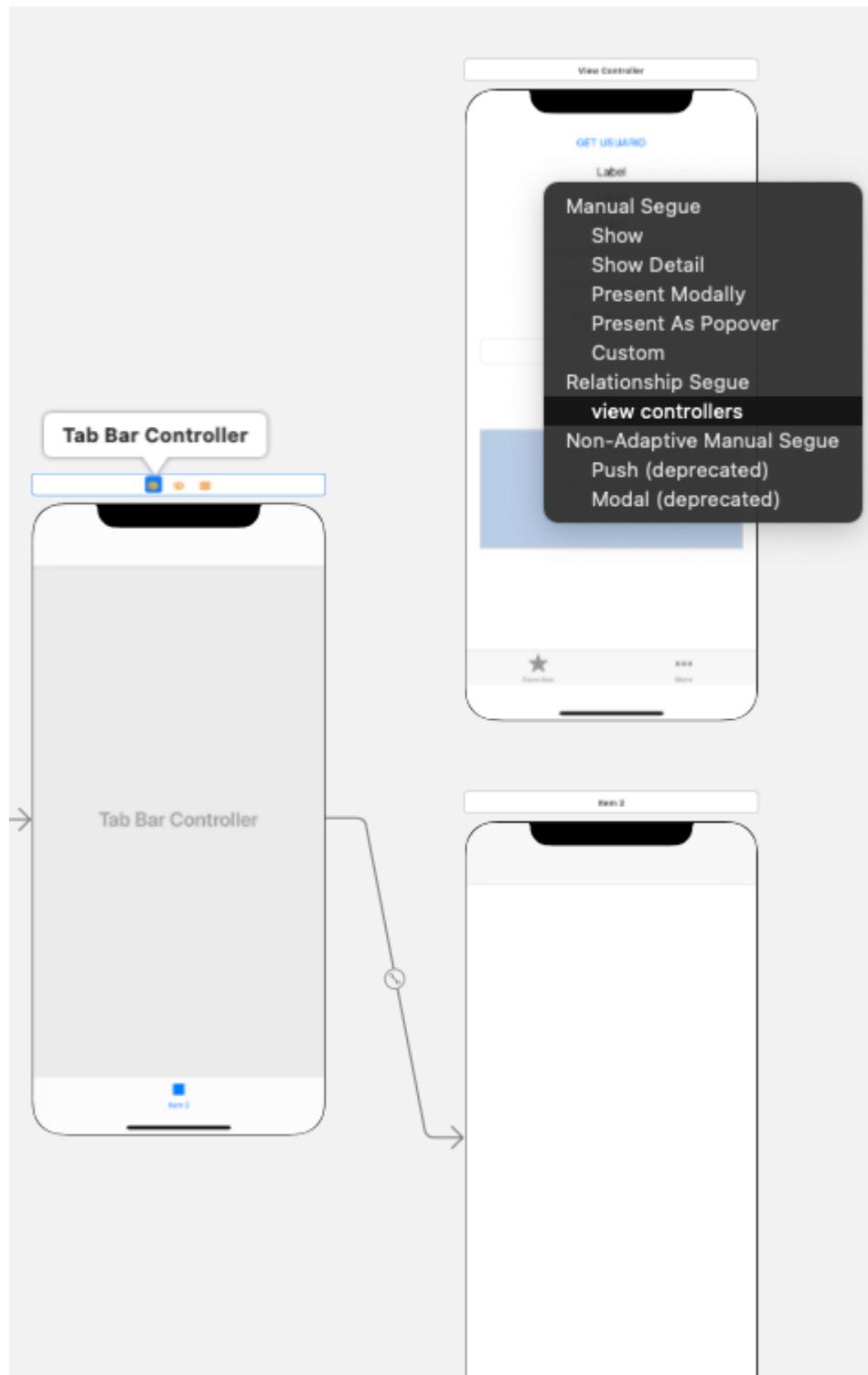


Para descargar imagenes en una app lo mejor es instalar la libreria kingfisher github. Para instalar una libreria lo que hay que hacer es coger todo el codigo de github, irme al link de la pagina y añadirlo a los packages añadiendo .git (punto git).

repository.

<https://github.com/onevcat/Kingfisher.git>

Para poner un tab bar inferior para que me cambie de pantalla segun pulse, tengo que poner un tab bar controller y luego, si quiero usar una pantalla mia propia, tengo que linkearla y darle a relationship segue -> view controllers.



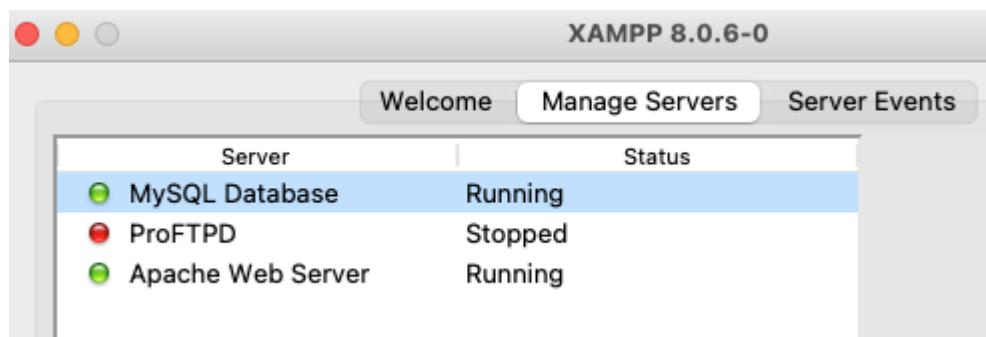
11/05/2021

Siempre que hago un tableview y no se ve, tengo que mirar si tengo que hacer un tableview.reloadData().

13/05/2021

la libreria que gestiona las bases de datos de php en android se llama volley. Tambien tengo que poner en el manifest el permiso para usar internet y tambien crear un xml de network-security-config en el que tengo que poner las paginas web seguras.

estamos usandoxampp a modo de base de datos. Si la instalo en casa y quiero usar mi ordenador a modo de base de datos para practicas y lo que sea tengo que abrir el programa y en la parte de Manage Servers tengo que activar el primer y el tercer database.



14/05/2021

estamos viendo notificaciones. Las notificaciones de Android tienen que ser para una API 26 en adelante y como no va a funcionar en todos los moviles, tengo que hacer dos notificaciones: unas para los moviles que tengan la API 26 o mas y otras para los que tengan las API 25 o menos.

17/05/2021

Cuando tengamos una app y queramos pedir permisos para lo que sea necesitamos el contextCompat.checkSelfPermission(contexto , Manifest.permission o Manifest.permission_group). Lo suyo es meter esa linea de codigo en una variable para poder ejecutarla y comprobar su estado.

```

//comprobaremos si tenemos los permisos correspondientes
private void concederPermisos(){

    int permisoNotificaciones = ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_NOTIFICATION_POLICY);

    //si permisoNotificaciones es igual a 0 es que SI me han dado permiso, y si es -1 es que NO me han dado permiso.

    if (permisoNotificaciones == PackageManager.PERMISSION_GRANTED){
        //Confirma que tengo los permisos del checkSelfPermission
    } else {
        //Confirma que NO tengo los permisos del checkSelfPermission
    }
}

```

En iOS las notificaciones son distintas: lo primero que tengo que hacer para hacer una notificación en Swift es utilizar una variable trigger.

```

func enviarNotificacion(){

    //primero creamos el trigger de la notificación
    let trigger = UNTimeIntervalNotificationTrigger(timeInterval: 10.0, repeats: false)

    //el segundo paso es crear el contenido de la notificación
    let notificación = UNMutableNotificationContent()

    notificación.title = "Título"
    notificación.subtitle = "Subtítulo"
    notificación.body = "¿Puedo creer lo que oigo? - dijo Ignatius."
    notificación.sound = UNNotificationSound.default

    //el tercer paso es crear un request que es lo que hará la notificación como tal
    let request = UNNotificationRequest(identifier: "prueba", content: notificación, trigger: trigger)

    //por último, añadimos el request al centro de notificaciones que es el que realiza la notificación
    UNUserNotificationCenter.current().removeAllDeliveredNotifications()
    UNUserNotificationCenter.current().add(request, completionHandler: {(error) in if let error = error {print("Se ha producido un error \n \(error)")}})

}

```

La constante notificación tiene que tener como mínimo un título. Este código es lo que hace una notificación pero no funciona si no le hemos dado permisos (hay que importar la librería UserNotifications). Luego, en el archivo App Delegate tengo que modificar este método para que me enseñe el alert.

```

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    UNUserNotificationCenter.current().requestAuthorization(options: [.alert, .sound]) { (confirmación, error) in
        if !confirmación {
            print("Permiso denegado por el usuario.")
        }
    }

    return true
}

```

Con este código lo que consigo es una notificación en segundo plano y solo cuando salgo de la app (le doy a enviar notificación y me salgo de la app, y entonces ya me sale).

Para que me salga como una notificación estándar necesito una extensión y un delegate del notificaciónCenter.

```

notificacion.sound = UNNotificationSound.default

//tenemos que declarar un delegado
UNUserNotificationCenter.current().delegate = self

//el tercer paso es crear un request que es lo que hara
let request = UNNotificationRequest(identifier: "prueba"
extension ViewController: UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter, willPresent notification: UNNotification,
        withCompletionHandler completionHandler: @escaping (UNNotificationPresentationOptions) -> Void) {
        completionHandler([.banner, .sound])
    }
}

```

18/05/2021

hoy estamos viendo como hacer notificaciones cloud usando firebase messaging.

para insertar un servicio personalizado tengo que crear una clase de java.

en el manifes

```

<service
    android:name=".MyFirebaseMessagingService"
    android:exported="false">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
</service>

```

En mi clase de java para ese servicio

```

package com.javierms.notificacionespush;

import com.google.firebase.messaging.FirebaseMessagingService;

public class MyFirebaseMessagingService extends FirebaseMessagingService {
}

```

si hago una notificacion push puedo:

- 1º - puedo crear una notificacion para todos los dispositivos que esten usando mi app
- 2º - puedo hacer una notificacion para un dispositivo en concreto usando un token porque quiero notificar a alguien en concreto de algo en concreto (que he actualizado algo, que me pague un servicio...)

Para que llegue la notificacion tengo que poner en el manifest el user permission -> internet (si no tengo internet no me llegara nada, pero si no pido permiso al usuario no tendre internet).

Para crear una notificacion especifica para un telefono en concreto, tengo que usar un token.

19/05/2021

el examen seran 20 preguntas tipo test de iOs y una practica que sera hacer un tab bar con 3 botones que llevarian a 3 pantallas:

- la primera pantalla es para introducir un establecimiento: meter una imagen, titulo, descripcion y un boton de añadir para guardar la informacion en un objeto tipo establecimiento{titulo: String, descripcion: String, latitud: Int, longitud: Int}.
- la segunda tiene un tableview con un array de establecimientos y tiene que mostrar los establecimientos basicos ademas de lo que yo cree.
- la tercera pantalla lo que hara sera llevarme a un mapa con map kit y mostrarme que establecimiento he pulsado en la pantalla 2 o todas las localizaciones que tengo en el array de establecimientos.

Esto es el codigo para conseguir que el usuario me de el permiso para enseñarle notificaciones y se pone en el app delegate.

```

func getPushNotificaciones(){
    UNUserNotificationCenter.current().getNotificationSettings { (settings) in
        switch settings.authorizationStatus{
            //el usuario tiene que registrar la notificacion
            case .notDetermined:
                self.getAutorizarNotificaciones()

            case .authorized:
                DispatchQueue.main.async {
                    UIApplication.shared.registerForRemoteNotifications()
                }
            case .denied:
                print("permiso denegado")

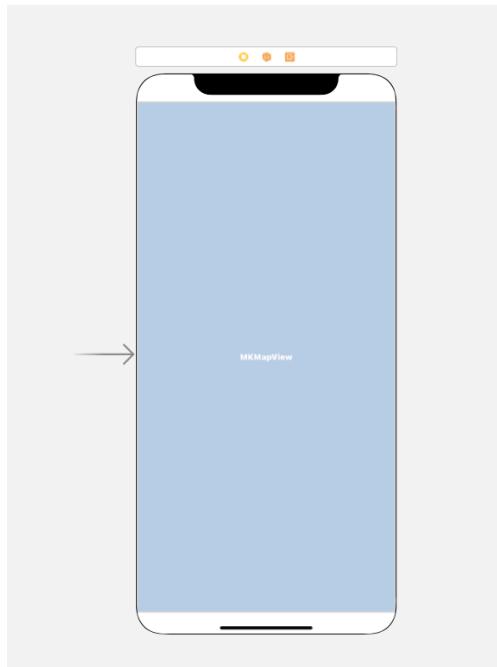
            default:
                break
        }
    }
}

func getAutorizarNotificaciones(){
    UNUserNotificationCenter.current().requestAuthorization(options: [.alert, .sound, .badge]) { (acceso, error) in
        if acceso {
            DispatchQueue.main.async {
                UIApplication.shared.isRegisteredForRemoteNotifications =
            }
        }
    }
}

```

en la practicaMapas2 vamos a ver como hacer casi todo lo del examen (localizacion y ruta).

Primero hago la pantalla del mapa poniendole el map kit y las constrain a 0:



Importo el mapkit, le pongo el mapa en outlet llamandole miMapa y le digo que estoy en getafe centro de emprendedores. Este codigo me localiza el sitio pero no me lo enseña porque me enseña el mapa de toda españa

```

import UIKit
import MapKit

class ViewController: UIViewController {

    @IBOutlet weak var miMapa: MKMapView!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.

        //para centrar la vista en un punto (centro de emprendedores)
        miMapa.centerCoordinate = CLLocationCoordinate2D.init(latitude: 40.29402174904825, longitude: -3.744329625609912)
    }
}

```

aqui le digo donde esta el metro Conservatorio y lo añado a mi mapa

```

//para sacar un punto punto

let conservatorio = MKPointAnnotation()
conservatorio.coordinate = CLLocationCoordinate2D.init(latitude: 40.2934549432367, longitude: -3.7461632225390833)
conservatorio.title = "metro Conservatorio"

//mostrar ese punto en el mapa
miMapa.addAnnotation(conservatorio)

```

hago una region para que el mapa se muestre una zona concreta cuando se abra

```

//mostrar ese punto en el mapa
miMapa.addAnnotation(conservatorio)

//aqui creo una region para que me muestre el mapa de cerca
let region = MKCoordinateRegion(center: conservatorio.coordinate, latitudinalMeters: 1000, longitudinalMeters: 1000)

//Mostrar la region que he declarado arriba
miMapa.setRegion(region, animated: true)

```

20/05/2021

Para crear un objeto recordamos que usamos un Swift File. Para crear un objeto de lugar lo que hacemos es crear el objeto y meterle sus atributos:

```

7
8 import Foundation
9 import MapKit
10
11
12 class cLugar {
13
14     var titulo:String
15     var descripcion: String
16     var latitud: Double
17     var longitud: Double
18
19     init(titulo: String, descripcion: String, latitud: Double, longitud: Double) {
20
21         self.titulo = titulo
22         self.descripcion = descripcion
23         self.latitud = latitud
24         self.longitud = longitud
25
26     }
27
28 }

```

Aqui creo un array de objetos de tipo cLugar y le añado los objetos que yo quiera.

```
//Aqui creo un array de objetos para llenarlo de lugares.
var listLugares: [cLugar]

//aqui empiezo a añadirle cosas
listLugares.append(cLugar(titulo: "Conservatorio", descripcion: "parada de metro", latitud: 40.29357404552661,
    longitud: -3.7458406896253402))
listLugares.append(cLugar(titulo: "Arroyo Culebro", descripcion: "parada de metro", latitud: 40.28949161921757,
    longitud: -3.755110102561311))
listLugares.append(cLugar(titulo: "Alonso Mendoza", descripcion: "parada de metro", latitud: 40.301297697419564,
    longitud: -3.7362197231815744))
```

Luego hago un bucle que los recorra y los añada.

```
//creo un bucle para que recorra el array
for lugar in listLugares{

    let punto = MKPointAnnotation()

    punto.title = lugar.titulo
    punto.subtitle = lugar.descripcion
    punto.coordinate = CLLocationCoordinate2D(latitude: lugar.latitud, longitude: lugar.longitud)

    miMapa.addAnnotation(punto)
}
```

Hago de nuevo el codigo para que me centre el mapa en el primer punto del array:

```
//aqui creo una region para que me muestre el mapa de cerca
let region = MKCoordinateRegion(center: CLLocationCoordinate2D(latitude: listLugares[0].latitud, longitude:
    listLugares[0].longitud), latitudinalMeters: 2500, longitudinalMeters: 2500)

//Mostrar la region que he declarado arriba
miMapa.setRegion(region, animated: true)
```

Para hacer una ruta que une puntos lo que necesito usar es MKDirections(). Primero hago una constante que se llama direcciones y luego hago el metodo

```
let direcciones = MKDirections()
```

IMPORTANTE: como voy a estar usando la localizacion del usuario tengo que ir al infoplist. y añadir estos permisos y esa descripcion.

Information Property List	Dictionary	(19 items)
Privacy - Location When In Use Usage Description	String	Allow location acces in order to use this app
Privacy - Location Always Usage Description	String	Allow location acces in order to use this app
Privacy - Location Always and When In Use Usage Description	String	Allow location acces in order to use this app

Para sacar la localizacion actual del usuario tengo que hacer un locationManager, importarle el CoreLocation y ponerle un delegado.

```

import CoreLocation

class ViewController: UIViewController, CLLocationManagerDelegate {

    @IBOutlet weak var miMapa: MKMapView!

    //Aqui creo un array de objetos para llenarlo de lugares.
    var listLugares: [cLugar] = []

    let direcciones = MKDirections()
    let manager = CLLocationManager()

    override func viewDidLoad() {
        super.viewDidLoad()

        //Aqui pongo a funcionar al location manager para que el telefono coja la localizacion del usuario
        manager.delegate = self
    }
}

```

luego le pongo unas cosas mas:

```

override func viewDidLoad() {
    super.viewDidLoad()

    //Aqui pongo a funcionar al location manager para que el telefono coja la localizacion del usuario
    manager.delegate = self

    //tengo que elegir el grado de precision a la hora de localizar
    manager.desiredAccuracy = kCLLocationAccuracyBest

    //solicito autorizacion al usuario para ver si me deja acceder a su localizacion
    manager.requestAlwaysAuthorization()

    //inicia la localizacion
    manager.startUpdatingLocation()
}

```

y luego la funcion que localiza al usuario en el telefono

```

func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
    print("Localización: \(locations)")
    print("Latitud: \(locations[0].coordinate.latitude)")
    print("Longitud: \(locations[0].coordinate.longitude)")

}

```

Para hacer rutas lo que tengo que hacer es usar el MKDirections y para ello necesito decirle donde empiezo, donde acabo y como voy (andando, en coche...)

```

//Aqui preparo la ruta y el medio de transporte
//Mis puntos de prueba
let p1 = MKPlacemark(coordinate: CLLocationCoordinate2D(latitude: listLugares[0].latitud, longitude: listLugares[0].latitud))
let p2 = MKPlacemark(coordinate: CLLocationCoordinate2D(latitude: listLugares[2].latitud, longitude: listLugares[2].latitud))

//crear direcciones y rutas
let request = MKDirections.Request()

//punto de origen, de destino y forma de desplazarse
request.source = MKMapItem(placemark: p1)
request.destination = MKMapItem(placemark: p2)
request.transportType = .transit

```

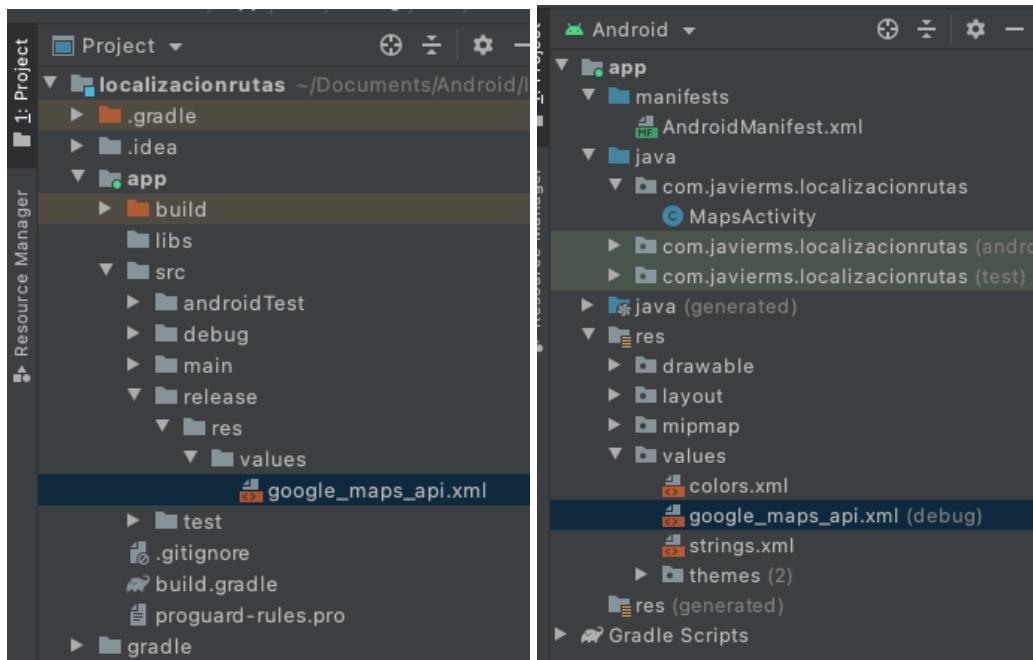
21/05/2021

Hoy usaremos la API de maps de google en android, así que lo primero que tenemos que hacer es coger los textos sagrados:

- <https://developers.google.com/maps/documentation?hl=es>
- <https://developers.google.com/maps/documentation/android-sdk/overview?hl=en>

Tengo que crear un proyecto nuevo y conseguir una clave de API en google cloud platform y poner en los dos xml de maps:

en la app normal y en el build gradle del proyecto



tambien tengo que poner estos permisos en el manifest:

```
<uses-permission android:name="android.hardware.location.gps"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

Aqui creo los puntos que quiero enseñar y le digo que me centre la camara en el conservatorio. Siempre que quiera mostrar un nuevo punto en el mapa tengo que crear una constante de tipo LatLng (un objeto Latitud - Longitud).

```
    }
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        //Objetos con coordenadas
        LatLng m1 = new LatLng( v: 40.293424241968474, v1: -3.7461354933791293);
        LatLng m2 = new LatLng( v: 40.28889069949901, v1: -3.7568993835911404);
        LatLng m3 = new LatLng( v: 40.3008866323001, v1: -3.7363067933057343);

        //añadir lugar al mapa
        mMap.addMarker(new MarkerOptions().position(m1).title("Conservatorio"));
        mMap.addMarker(new MarkerOptions().position(m2).title("Arroyo Culebro"));
        mMap.addMarker(new MarkerOptions().position(m3).title("Alonso Mendoza"));

        //Aqui centro la camara en el conservatorio
        mMap.moveCamera(CameraUpdateFactory.newLatLng(m1));
    }
```

Aqui hago zoom para que despliegue la camara lo suficientemente cerca para que se vean los tres puntos que he declarado arriba:

```
//Aqui hago que la camara haga zoom al punto que le digo
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(m1, v: 14));
```

Y esto para cambiar el tipo de mapa

```
//Cambiar el estilo del mapa (politico, geografico o hibrido)
mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

Este es el metodo para pedir permiso al usuario para que me deje acceder en su ubicacion, y este metodo lo tengo que meter en el onCreate para que lo ejecute nada mas entrar a la app:

```
//Funcion para preguntar al usuario si concede permiso
public void getPermisoLocalizacion(){

    int permiso = ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION);

    if (permiso == PackageManager.PERMISSION_DENIED){
        if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this,Manifest.permission.ACCESS_FINE_LOCATION)){
            } else {
                ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, requestCode: 1);
            }
    }
}
```

Y aqui accedo a su ubicacion. IMPORTANTE: el metodo setMyLocationEnabled me va a dar un error nada mas ponerlo porque tiene que hacer una comprobacion de seguridad para saber si tengo permiso para acceder a ello. Si le doy a la bombilla roja del error me saca el metodo entero.

```
//Acceso a la localizacion del usuario - PRIMERO LE TENGO QUE DAR PERMISO
if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
    && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
    // TODO: Consider calling
    //     ActivityCompat#requestPermissions
    // here to request the missing permissions, and then overriding
    //     public void onRequestPermissionsResult(int requestCode, String[] permissions,
    //                                         int[] grantResults)
    // to handle the case where the user grants the permission. See the documentation
    // for ActivityCompat#requestPermissions for more details.
    return;
}
mMap.setMyLocationEnabled(true);
```

Para añadir un icono al mapa de maps lo que tengo que hacer es un icon con el .icon() pero ese metodo lo que me pide es un bitmap:

```
//Aqui lo que hago es crear una imagen para ponerla en el mapa
int height = 90;
int width = 130;
BitmapDrawable bitmapdraw = (BitmapDrawable) getBaseContext().getDrawable( R.drawable.metro);

//Aqui hago ya el icono
Bitmap icon = bitmapdraw.getBitmap();
Bitmap iconmin = Bitmap.createScaledBitmap(icon, width, height, filter: true);

//añadir lugar al mapa
mMap.addMarker(new MarkerOptions().position(m1).title("Conservatorio").icon(BitmapDescriptorFactory.fromBitmap(iconmin)));
```

24/05/2021

Hoy estamos con SwiftUI.

Cuando uso el Swift UI tengo que tener cuidado con los elementos porque si no le digo que embeba los elementos, me va a crear una pantalla por elemento.

```
import SwiftUI

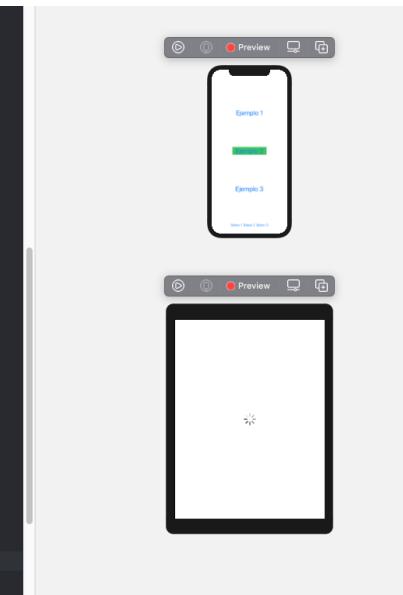
struct ContentView: View {
    var body: some View {
        VStack {
            Text("Página 1")
                .font(.largeTitle)
                .foregroundColor(.blue)
                .padding(.horizontal,20.0)

            Text("Página 2")
                .font(.largeTitle)
                .foregroundColor(.blue)
                .padding(.horizontal,16.0)
                .background(Color.green)

            Text("Página 3")
                .font(.largeTitle)
                .foregroundColor(.blue)
                .padding(.horizontal,12.0)
        }
    }
}
```

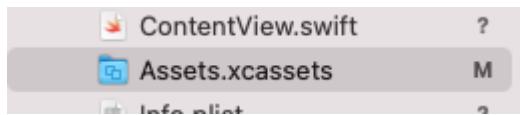
este código me crea 3 elementos en la misma pantalla y gracias al VStack me lo deja colocado como si fuera un vertical layout.

```
39         Button(action: {}, label: {
40             Text("Botón 1")
41         })
42
43         Button(action: {}, label: {
44             Text("Botón 2")
45         })
46
47         Button(action: {}, label: {
48             Text("Botón 3")
49         })
50     }
51
52 }
53
54
55
56
57
58
59
60
61 }
62 }
63 struct ContentView_Previews: PreviewProvider {
64     static var previews: some View {
65         ContentView()
66         ContentView().previewDevice("iPad")
67     }
68 }
69
```



Con el `ContentView().previewDevice` lo que consigo es poder ver mi app en varias pantallas de varios dispositivos a la vez.

Para añadir imagenes tengo que hacerlo desde la carpeta assets



Como en SwiftUI no tengo todos los componentes (no tengo el mapa de google maps, por ejemplo), asi que tengo que importar elementos del StoryBoard. Lo primero que tengo que hacer es importar la libreria que quiero usar, luego le pongo a la estructura que es un representante del elemento que me traigo, luego hago la funcion para usar el elemento y luego hago una funcion para que le diga a la estructura que use ese representante

```
import SwiftUI
import MapKit

/*struct mapView: View {
    var body: some View {
        Text("Hello, World!")
    }
}*/
```

```
struct mapView: UIViewRepresentable {

    //Componente del UIKit del antiguo sistema que quiero llamar
    func makeUIView(context: Context) -> MKMapView {
        MKMapView(frame: .zero)
    }

    //Actualizar y devolver esta vista a la estructura
    func updateUIView(_ uiView: MKMapView, context: Context) {
    }

}
```

```
//Actualizar y devolver esta vista a la estructura
func updateUIView(_ uiView: MKMapView, context: Context) {

    let coordenadasInicio = CLLocationCoordinate2D(latitude: 40, longitude: -3)

    let span = MKCoordinateSpan(latitudeDelta: 15, longitudeDelta: 15)

    let region = MKCoordinateRegion(center: coordenadasInicio, span: span)

    let punto = MKPointAnnotation()

    punto.coordinate = coordenadasInicio

    uiView.addAnnotation(punto)

    uiView.setRegion(region, animated: true)

}
```

25/05/2021

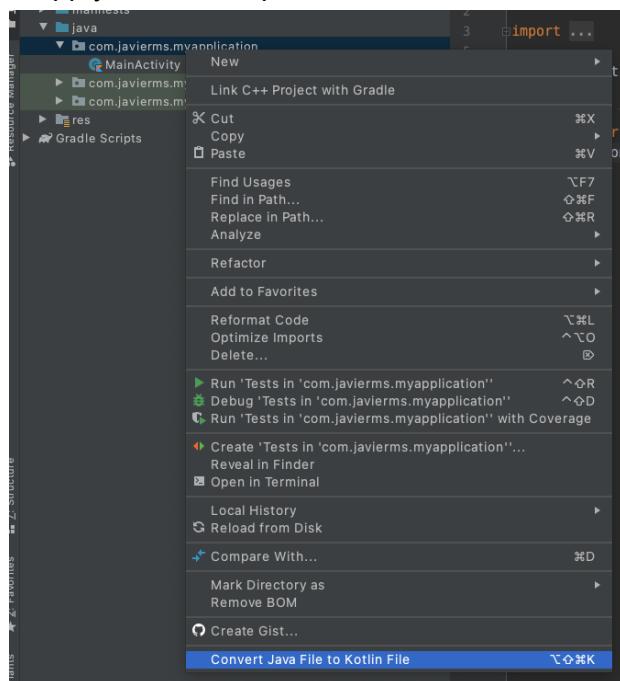
Para hacer una celda personalizada en SwiftUI es la celda, que va a ser un archivo que se llame celda y sera de tipo SwiftUI View.

De SwiftUI apenas hay nada, pero si queremos mirar algo podemos mirar moureDev.

La diferencia entre una estructura y un objeto es que el objeto o clase tiene atributos y funciones, pero la estructura no, la estructura solo tiene lo que yo quiera que tenga. Al tener menos datos obligatorios, la estructura es mejor para traer traer datos de la base de datos, asi no me traigo datos innecesarios y no me gastos recursos en cosas que no me van a servir.

para el examen de certificacion es importante saber:

- kotlin: ¿cuando se creo kotlin? En julio del 2011.
- pregunta trampa: cuando kotlin empezo a funcionar fue en febrero de 2012 cuando compilo por primera vez mas rapido que java.
- ventajas de kotlin (son 6): performance, kotlin compila mas rapido que java; es facil de entender e intuitivo; facil integracion con android studio; interoperabilidad, puede operar facilmente al 100% con todo java; se escribe menos codigo; kotlin no tienes que declararlo todo, puedes declarar las cosas de forma nula.
- los estados de las activities
- para pasar de java a kotlin y viceversa tienes que dale click derecho a la carpeta o a la app y darle a la opcion del fondo del todo.



- java funciona gracias a una maquina virtual que tiene instalada el ordenador y kotlin no necesita esa maquina virtual.
- instalacion del entorno de desarrollo llamado IntelliJ Idea y tengo que saber donde esta la parte de los SDK para poder instalarlos.

- a la hora de declarar las cosas en kotlin las variables se declaran con var y las constantes con val y no hay que decirle que tipo de dato es, kotlin lo infiere.

```
//constantes
val mes = "Marzo"

//variables
var num1 = 2
```

- en tiempo de ejecucion las constantes NO SE PUEDEN CAMBIAR DE TIPO DE DATO pero las variables SI.
- Aqui tenemos una funcion en kotlin con un parametro llamado numero de tipo Int y que devuelve un valor Int.

```
private fun declaracion(numero: Int): Int {

    var num1 = numero

    return num1
}
```

26/05/2021

Mirar como pedir permisos en iOs.

Para recoger informacion de los edittext de la pantalla y crear un nuevo objeto con esos atributos, lo primero que tengo que hacer un delegado del tipo UITextFieldDelegate y añadirle las funciones textFieldShouldReturn y dentro resignFirstResponder para que se esconda el teclado cuando le de al intro y la funcion textFieldDidEndEditing para recoger los datos de los campos de texto. Al inicio de la app, ademas, tengo que añadirle un objeto vacio que va a ser donde guarde los atributos y los envie a otra pantalla.

```
class cHuracan {

    private var nombre: String
    private var anio: String
    private var kmh: Double
    private var id: Int

    init(nombre: String, anio: String, kmh: Double, id: Int){
        self.nombre = nombre
        self.anio = anio
        self.kmh = kmh
        self.id = id
    }

    func getNombre() -> String { return self.nombre }
    func getAnio() -> String { return anio }
    func getKmh() -> Double { return kmh }
    func getId() -> Int { return id }

    func setNombre(nombre: String) { self.nombre = nombre }
    func setAnio(anio: String) { self.anio = anio }
    func setKmh(kmh: Double) { self.kmh = kmh }
    func setId(id: Int) { self.id = id }

}
```

```

extension ViewController: UITextFieldDelegate {

    func textFieldShouldReturn(_ textField: UITextField) -> Bool {
        //esta funcion lo que hace es cerrar el teclado al pulsar el enter
        textField.resignFirstResponder()
    }

    //para guardar el dato cuando el usuario deje de editar
    func textFieldDidEndEditing(_ textField: UITextField) {
        altaHuracan.setNombre(nombre: String(etxtNombre.text!))
        altaHuracan.setAnio(anio: String(etxtAño.text!))
        altaHuracan.setKmh(kmh: Double(etxtKmh.text!)!)
        altaHuracan.setId(id: Int(etxtID.text!)!)

    }

}

```

Aqui hay un segue completo con prepare y perform

```

@IBAction func btnRegistrarAction(_ sender: Any) {
    performSegue(withIdentifier: "haciaRecoger", sender: self)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "haciaRecoger"{
        if let destino = segue.destination as? recogerHuracan{
            destino.recogerHuracan = altaHuracan
        }
    }
}

```

Para llevar a cabo el prepare for segue y enviar un objeto a otra pantalla, tengo que crear una variable o un contenedor que va a contener el objeto que le mando. En el ejemplo de arriba vemos que destino.recogerHuracan = altaHuracan, lo cual significa:

- destino es igual al segue.destination, que es hacia donde va el segue
- recogerHuracan es la variable que he creado en la pantalla de destino para que se convierta en el objeto que estoy enviando
- altaHuracan es el objeto que estoy enviando, cuyos datos cojo de los etxt.

28/02/2021

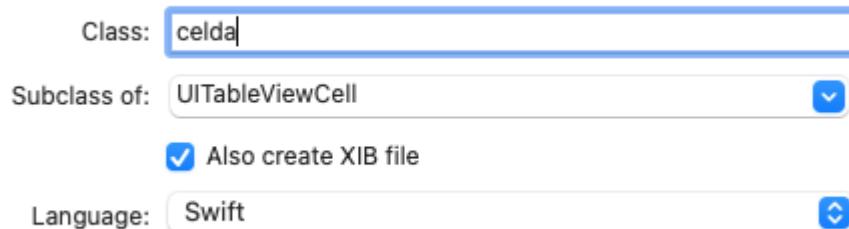
Hay dos tipos de celdas en los tableview: las del sistema o las personalizadas. Las celdas simples solo tienen un label, por lo que no son lo mejor si quiero que me muestre objetos o cosas con multiples campos.

las celdas simples del sistema se hacen asi:

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    //Celda simple en la que solo viene un label  
    let celda = UITableViewCell(style: .default, reuseIdentifier: "miCelda")  
  
    celda.textLabel?.text = recogerHuracan[indexPath.row].getNombre()  
  
    return celda  
  
}
```

Para sacar celdas personalizadas primero tengo que crear el archivo de la celda y luego gestionarlo desde el tableview. La celda se crea asi:

Creo un Cocoa touch class de tipo UITableViewCell y darle el clicl al XIB file, que es la parte grafica:



Luego a la celda tengo que ponerle un idenfiticador



Luego tengo que registrar mi celda personalizada en la pantalla que tiene el tableview.

```
lvArrayHuracanes.register(UINib(nibName: "celda", bundle: nil), forCellReuseIdentifier: "miCelda")
```

En el archivo de la celda tengo esto, solo tengo declarados los outlets

```
import UIKit

class celda: UITableViewCell {

    @IBOutlet weak var txtNombreCelda: UILabel!
    @IBOutlet weak var txtAnioCelda: UILabel!
    @IBOutlet weak var txtKMHcelda: UILabel!
    @IBOutlet weak var txtIDCelda: UILabel!

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code

        txtNombreCelda.text = ""
        txtAnioCelda.text = ""
        txtKMHcelda.text = ""
        txtIDCelda.text = ""
    }

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }
}
```

Y en la extension del datasource que me rellena la tabla tengo esto para usar la celda personalizada.

```
extension recogerHuracan: UITableViewDataSource, UITableViewDelegate {

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return recogerHuracan.count
    }

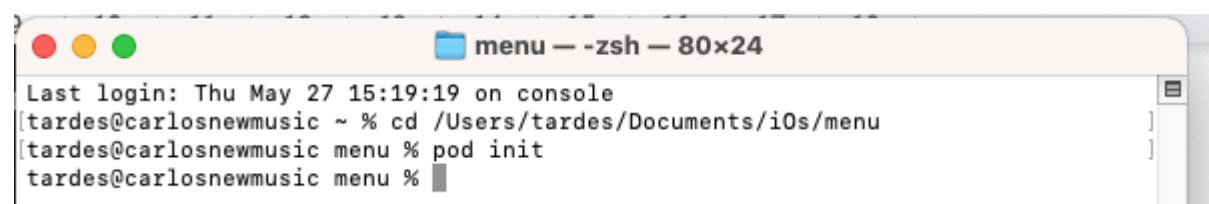
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        //Celda personalizada
        let celda = tableView.dequeueReusableCell(withIdentifier: "micelda", for: indexPath) as? celda

        let stringKMH = String(recogerHuracan[indexPath.row].getKmh())
        let stringID = String(recogerHuracan[indexPath.row].getId())

        celda?.txtNombreCelda.text = recogerHuracan[indexPath.row].getNombre()
        celda?.txtAnioCelda.text = recogerHuracan[indexPath.row].getAnio()
        celda?.txtKMHcelda.text = stringKMH
        celda?.txtIDCelda.text = stringID

        return celda!
    }
}
```

Para hacer un menu desplegable en iOs lo primero que tengo que hacer es meterle un archivo pod en el terminal. Abro el terminal y le pongo:



Y luego el pod lo abro con un editor de texto y le añado simplemente pod 'SideMenu' y luego vuelvo al terminar y pongo pod install. Despues de instalar un pod, si lo he hecho, se me creara un nuevo archivo de proyecto pero en lugar de ser azul sera blanco. Tengo que cerrar el proyecto azul y abrir de nuevo el xcode con el archivo blanco.

En el archivo en blanco le importo la libreria SideMenu y le escribo ese codigo para sacar un menu lateral (hay que fijarse que la funcion que sacar el menu, el present, se lo he puesto a un bar button item). Con esto, deberia salir un menu lateral vacio.

```
import UIKit
import SideMenu

class ViewController: UIViewController {

    var menu: SideMenuNavigationController?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.

        menu = SideMenuNavigationController(rootViewController: UIViewController())
    }

    @IBAction func btnAbrirMenuAction(_ sender: Any) {
        present(menu!, animated: true)
    }
}
```

28/05/2021

para instalar cocoa pods tengo que ir a <https://cocoapods.org> y seguir esas instrucciones para que se me baje un archivo para tener acceso a ciertas librerias, frameworks y cosas. En la pagina me vienen las instrucciones para instalarlo. Siempre que añada cualquier cosa al archivo pod, tengo que ir al terminal y ponerle pod install para que se me baje las librerias que yo le haya metido al archivo pod.

Para crearle los botones, lo que tengo que hacer es crear la clase MenuLista, luego lo que hago es crear un array con los botones que quiero hacer, luego creo una celda por cada boton y luego registro esa celda para que el menu lo pueda usar.

```
class MenuLista: UITableViewController{

    var botones = ["Inicio", "Ayuda", "Servicios", "Configuracion"]

    override func viewDidLoad() {
        super.viewDidLoad()

        tableView.register(UITableViewCell.self, forCellReuseIdentifier: "miCelda")
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return botones.count
    }

    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let celda = tableView.dequeueReusableCell(withIdentifier: "miCelda", for: indexPath)

        celda.textLabel?.text = botones[indexPath.row]

        return celda
    }

}
```

```
override func viewDidLoad() {
    super.viewDidLoad()

    menu = SideMenuNavigationController(rootViewController: MenuLista())

    menu?.leftSide = true
}
```

Para que el menu se despliegue deslizando el dedo:

```
//gestos drag menu
SideMenuManager.default.leftMenuNavigationController = menu

SideMenuManager.default.addScreenEdgePanGesturesToPresent(toView: self.view)
```

Para modificar el color de fondo del menu. Si quiero modificar el color de fondo de la celda lo tengo que hacer desde la funcion que saca la celda, el cellForRowAt.

```
override func viewDidLoad() {
    super.viewDidLoad()

    tableView.backgroundColor = colorMenu

    tableView.register(UITableViewCell.self, forCellReuseIdentifier: "miCelda")
}
```

