

RESOLUCION DE ERRORES

```
41 fun AddView(  
44     cronosVM: CronosViewModel  
45 ){  
46     Scaffold(  
47         topBar = {  
48             //clase de material que crea una barra en la parte superior  
49             CenterAlignedTopAppBar(  
50                 title = { MainTitle(title = "Add Crono") },  
51                 colors = TopAppBarDefaults.centerAlignedTopAppBarColors(  
52                     containerColor = MaterialTheme.colorScheme.primary  
53                 ),  
54                 navigationIcon = {  
55                     //definido en la clase Buttons.kt  
56                     MainIconButton(icon = Icons.Default.ArrowBack) {  
57                         navController.popBackStack()  
58                     }  
59                 }  
60             )  
61         }  
62     ) {  
63         ContentAddView(it, navController, cronometroVM, cronosVM )  
64     }  
65 }
```

EN LA CLASE ADDVIEW EL PRIMER ERROR QUE NOS DA ES QUE LE FALTA UN } EN LA LINEA 58.

EL SIGUIENTE ERROR QUE ME DIO FUE AL ABRIRLO LE FALTABA IMPORTAR LAS DEPENDENCIAS DE CIRCLEBUTTON

```
41 fun AddView(  
44     cronosVM: CronosViewModel  
45 ){  
46     Scaffold(  
47         topBar = {  
48             //clase de material que crea una barra en la parte superior  
49             CenterAlignedTopAppBar(  
50                 title = { MainTitle(title = "Add Crono") },  
51                 colors = TopAppBarDefaults.centerAlignedTopAppBarColors(  
52                     containerColor = MaterialTheme.colorScheme.primary  
53                 ),  
54                 navigationIcon = {  
55                     //definido en la clase Buttons.kt  
56                     MainIconButton(icon = Icons.Default.ArrowBack) {  
57                         navController.popBackStack()  
58                     }  
59                 }  
60             )  
61         }  
62     ) {  
63         ContentAddView(it, navController, cronometroVM, cronosVM )  
64     }  
65 }
```

IMPORT COM.EXAMPLE.ROOMCRONOAPP.COMPONENT.CIRCLEBUTTON

ESTO SE ENCUENTRA EN LA CLASE **ADDVIEW**

SE DEBE DE COLOCAR EL } Y EL IMPORT

COM.EXAMPLE.ROOMCRONOAPP.COMPONENT.CIRCLEBUTTON EN EL
ÁREA DONDE SE IMPORTA LAS DEPENDENCIAS DENTRO DE LA CLASE

```
Row(horizontalArrangement = Arrangement.Center,
    modifier = Modifier.padding(vertical = 16.dp)) {
    //iniciar
    CircleButton(icon = painterResource(id = R.drawable.play), enabled = !state.cronometroActivo) {
        cronometroVM.iniciar()
    }
    //pausar
    CircleButton(icon = painterResource(id = R.drawable.pause), enabled = state.cronometroActivo) {
        cronometroVM.pausar()
    }
    //detener
    CircleButton(icon = painterResource(id = R.drawable.stop), enabled = !state.cronometroActivo) {
        cronometroVM.detener()
    }
    //mostrar TextField
    CircleButton(icon = painterResource(id = R.drawable.guardar), enabled = state.showSaveButton) {
        cronometroVM.showTextField()
    }
}
```

```

package com.example.roomcronoapp.model

import ...

//una entidad representa una tabla en base de datos
@Entity(tableName = "cronos")
data class Cronos(
    //definimos una llave primaria
    @PrimaryKey(autoGenerate = true)
    id: Long = 0,
    //column info representa un campo en nuestra tabla
    @ColumnInfo(name = "title")
    title: String,
    @ColumnInfo(name = "crono")
    crono: Long
)

```

EN LA CLASE CRONOS

ERROR: ENTITIES AND POJOS MUST HAVE A USABLE PUBLIC CONSTRUCTOR. YOU CAN HAVE AN EMPTY CONSTRUCTOR OR A CONSTRUCTOR WHOSE PARAMETERS MATCH THE FIELDS (BY NAME AND TYPE).

PUBLIC FINAL CLASS CRONOS {

^

TRIED THE FOLLOWING CONSTRUCTORS BUT THEY FAILED TO MATCH:

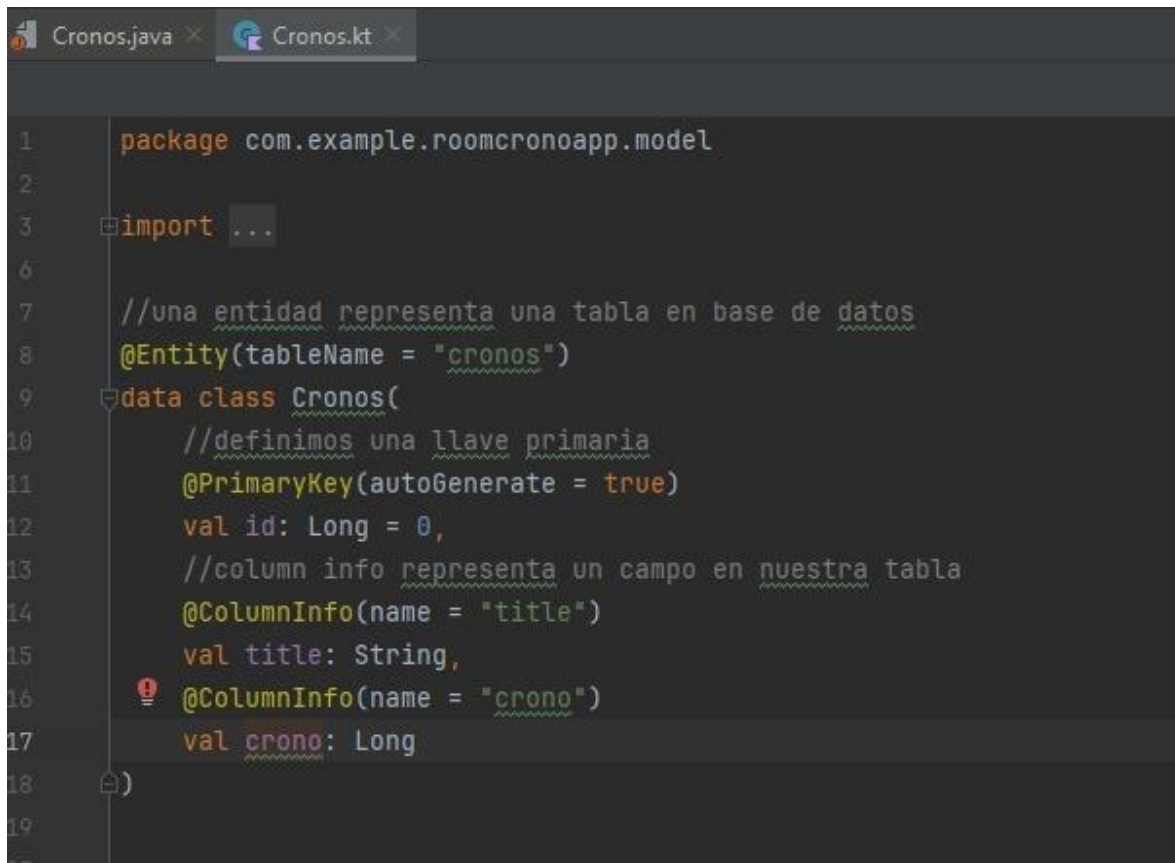
CRONOS(LONG, JAVA.LANG.STRING, LONG) -> [PARAM:ID -> MATCHED FIELD:UNMATCHED, PARAM:TITLE -> MATCHED FIELD:UNMATCHED, PARAM:CRONO -> MATCHED FIELD:UNMATCHED]

ESTE ERROR ES PORQUE EN EL MODELO CRONOS

MODEL.CRONOS

NO ESTÁ DEFINIDO CORRECTAMENTE LAS VARIABLES DATA CLASS CRONOS

CORREGIDO



```
1 package com.example.roomcronoapp.model
2
3 import ...
4
5
6
7 //una entidad representa una tabla en base de datos
8 @Entity(tableName = "cronos")
9 data class Cronos(
10     //definimos una llave primaria
11     @PrimaryKey(autoGenerate = true)
12     val id: Long = 0,
13     //column info representa un campo en nuestra tabla
14     @ColumnInfo(name = "title")
15     val title: String,
16     @ColumnInfo(name = "crono")
17     val crono: Long
18 )
19
20
```

LOS OTROS ERRORES QUE DA SON DOS ERRORES

NO VALUE PASSED FOR PARAMETER 'CRONOMETROVM'

NO VALUE PASSED FOR PARAMETER 'CRONOSVM'

Y ESTO OCURRE PORQUE EN LA CLASE NAVMANAGER() SE RECIBEN ESTOS DOS PARAMETROS QUE SON DEL VIEW MODEL CRONOMETROVM CRONOSVM Y PARA SOLUCIONARLO SE DEBEN DE ENVIAR

```
import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            RoomcronsappTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    NavManager()
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
    )
}
```

```

import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        // Aquí obtienes los ViewModels
        val cronometroVM: CronometroViewModel by viewModels()
        val cronosVM: CronosViewModel by viewModels()

        setContent {
            RoomcronoappTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    NavManager(cronometroVM, cronosVM)
                }
            }
        }
    }
}

```

SE ARREGLA AGREGANDO

// AQUÍ LOS VIEWMODELS

VAL CRONOMETROVM: CRONOMETROVIEWMODEL BY VIEWMODELS()

VAL CRONOSVM: CRONOSVIEWMODEL BY VIEWMODELS()

Y MODIFICANDO NAVMANAGER(CRONOMETROVM,CRONOSVM)

```

import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        // Aquí obtienes los ViewModels
        val cronometroVM: CronometroViewModel by viewModels()
        val cronosVM: CronosViewModel by viewModels()

        setContent {
            RoomcronoappTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    NavManager(cronometroVM, cronosVM)
                }
            }
        }
    }
}

```

EN LA CARPETA COMPONENT EN EL ARCHIVO BODYCOMPONENTS HAY ERROR DONDE DICE PADDING ESTO DEBE DE IR EN MINUSCULA

OTRO ERROR ES

```

text(text = crono, fontSize = 20.sp)
}
//una linea para diferenciar entre cada elemento
Divider(
    modifier = Modifier
        .fillmaxWidth()
        .height(1.dp),
    color = MaterialTheme.colorScheme.primary
)
}

```

EN LA CARPETA COMPONENT EN EL ARCHIVO BODYCOMPONENTS HAY ERROR DONDE DICE FILLMAXWIDTH ESTO DEBE DE IR ESCRITO ASI FILLMAXWIDTH

OTRO ERROR ES

```

Button(
  onClick = onClick, //asignamos la funcion recibida
  shape = CircleShape, //agregar forma redondeada
  contentPadding = PaddingValues(8.dp),
  enabled = enabled, //por defecto estara deshabilitado
  Modifier = Modifier.padding(horizontal = 15.dp)
) {
  Icon(

```

EN LA CARPETA COMPONENT EN EL ARCHIVO BUTTONS HAY ERROR
DONDE DICE MODIFIER ESTO DEBE DE IR ESCRITO EN MINÚSCULA SOLO
EL MODIFIER=

OTRO ERROR

```

//consultamos en base de datos el crono por ID
LaunchedEffect(Unit){
  cronometroVM.getCronoById(id)
}

Column(
  Modifier = Modifier
    .padding(it)
    .padding(top = 30.dp)
    .fillMaxSize(),
  horizontalAlignment = Alignment.CenterHorizontally
) {

  Text(
    text = formatTiempo(cronometroVM.tiempo),
    fontSize = 50.sp

```

EN LA CARPETA VIEWS EN EDITHVIEW ESTA MAL ESCRITO MODIFIER DEBE
DE IR EN MINUSCULA MODIFIER =

OTRO ERROR

```

@Composable
fun HomeView(cronosVM: CronosViewModel){
    Scaffold(
        topBar = {
            //clase de material que crea una barra en la parte superior
            CenterAlignedTopAppBar(
                title = { MainTitle(title = "Crono App") },
                colors = TopAppBarDefaults.centerAlignedTopAppBarColors(
                    containerColor = MaterialTheme.colorScheme.primary
                ))
        },
        floatingActionButton = {
            //componente creado en la clase Botones
            FloatingActionButton {
                navController.navigate("AddView")
            }
        }
    ) {
        ContentHomeView(it, navController, cronosVM )
    }
}

```

EN LA CARPETA VIEW EN EL ARCHIVO HOMEVIEW ESTA EL ERROR NAVCONTROLLER ESTE ERROR SE RESUELVE DECLARANDO LA VARIABLE COMO PARAMETRO

FUN HOMEVIEW(NAVCONTROLLER: NAVCONTROLLER,CRONOSVM: CRONOSVIEWMODEL){

OTRO ERROR ES

```
import com.example.roomcronsapp.viewModels.CronosViewModel
import com.example.roomcronsapp.views.AddView
import com.example.roomcronsapp.views.EditView
import com.example.roomcronsapp.views.HomeView

@Composable
fun NavManager(cronometroVM: CronometroViewModel, cronosVM: CronosViewModel){
    // Crear el NavController

    NavHost(navController = navController, startDestination = "Home"){
        composable(route: "Home"){
            HomeView(navController, cronosVM)
        }
        composable(route: "AddView"){
            AddView(navController, cronometroVM, cronosVM)
        }
        composable(route: "EditView/{id}", arguments = listOf(
            navArgument(name: "id") { type = NavType.LongType }
        )){
            val id = it.arguments?.getLong(key: "id") ?: 0
            EditView(navController, cronometroVM, cronosVM, id)
        }
    }
}
```

EN LA CARPETA NAVIGATION EN EL ARCHIVO NAVIGATION.KT NO ESTA DECLARADA LA VARIABLE NAVCONTROLLER SE DEBE DE DECLARAR COMO

VAL NAVCONTROLLER = REMEMBERNAVCONTROLLER()

```
//agregar el cronosVM para enviarlo al contentHomeView
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun HomeView(navController: NavController, cronosVM: CronosViewModel){
    // Crear el NavController
    Scaffold(
        topBar = {
            //clase de material que crea una barra en la parte superior
            CenterAlignedTopAppBar(
                title = { MainTitle(title = "Crono App") },
                colors = TopAppBarDefaults.centerAlignedTopAppBarColors(
                    containerColor = MaterialTheme.colorScheme.primary
                )
            ),
        floatingActionButton = {
```

Y RESUELTO SE MIRA ASI

```
import com.example.roomcronsapp.viewModels.CronosViewModel
import com.example.roomcronsapp.views.AddView
import com.example.roomcronsapp.views.EditView
import com.example.roomcronsapp.views.HomeView

@Composable
fun NavManager(cronometroVM: CronometroViewModel, cronosVM: CronosViewModel){
    // Crear el NavController
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination = "Home"){
        composable(route = "Home"){
            HomeView(navController, cronosVM)
        }
        composable(route = "AddView"){
            AddView(navController, cronometroVM, cronosVM)
        }
        composable(route = "EditView/{id}", arguments = listOf(
            navArgument(name = "id") { type = NavType.LongType }
        )){
            val id = it.arguments?.getLong(key = "id") ?: 0
            EditView(navController, cronometroVM, cronosVM, id)
        }
    }
}
```

Y EL ULTIMO ERROR ES ESTE

```
class CronometroViewModel @Inject constructor(private val repository: CronosRepository) : ViewModel() {
    // Estado del cronometro
    @State private var cronometroActivo = false
    // Ejecuta el cronometro
    fun cronos() {
        if (state.cronometroActivo) {
            cronoJob?.cancel()
            // Ejecutamos una corrutina para incrementar el tiempo
            // Esto hace la funcion del cronometro
            cronoJob = viewModelScope.launch {
                while (true) {
                    delay(timeMillis = 1000)
                    // tiempo += 1000
                }
            }
        } else {
            cronoJob?.cancel()
        }
    }

    fun getCronoById(id: Long) {
        viewModelScope.launch(Dispatchers.IO) {
            repository.getCronoById(id).collect { item ->
            }
        }
    }
}
```

PARA QUE CORRA EL TIEMPO DEBEMOS DE QUITAR COMO COMENTARIO EL `//TIEMPO+=1000` EN LA CARPETA VIEWMODELS EN EL ARCHIVO CRONOMETROVIEWMODEL EN LA LINEA 83 MAS O MENOS


Y RESULTO SE MIRA ASI

```
class CronometroViewModel @Inject constructor (private val repository: CronosRepository) {  
    //ejecuta el cronometro  
    fun cronos(){  
        if(state.cronometroActivo){  
            cronoJob?.cancel()  
            //ejecutamos una corrutina para incrementar el tiempo  
            //esto hace la funcion del cronometro  
            cronoJob = viewModelScope.launch {  
                while (true){  
                    delay( timeMillis: 1000)  
                    tiempo += 1000  
                }  
            }  
        }else{  
            cronoJob?.cancel()  
        }  
    }  
  
    fun getCronoById(id:Long){  
        viewModelScope.launch(Dispatchers.IO) {  
            repository.getCronoById(id).collect { item ->  
                //item  
            }  
        }  
    }  
}
```


Y SIN ERRORES SON QUEDARIA ASI

Crono App

prueba 1

 00:00:13

prueba 2

 00:00:09





EDIT CRONO

00:00:06



Title


prueba 1

Editar




Crono App

prueba 1

 00:00:06


prueba 2

 00:00:09



Crono App

prueba 1

 00:00:06





Add Crono

00:00:06



titulo

prueba 1

Guardar

