

**Carpeta VistaEmpleados**

**EmpleadoScreen.kt**  
package com.example.mvvmprueba.ui.VistaEmpleado

import androidx.compose.foundation.Image

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.Button

import androidx.compose.material3.ButtonDefaults

import androidx.compose.material3.CircularProgressIndicator

import androidx.compose.material3.Text

import androidx.compose.runtime.\*

import androidx.compose.runtime.livedata.observeAsState

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import kotlinx.coroutines.launch

import com.example.mvvmprueba.R

@Composable

fun EmpleadoScreen(viewModel: EmpleadoViewModel) {

Box(

Modifier

.*fillMaxSize*()

.*background*(*Color*(0xFF1c4c96))

) **{**

Empleado(Modifier.*align*(Alignment.Center), viewModel)

**}**

}

@Composable

fun Empleado(modifier: Modifier, viewModel: EmpleadoViewModel) {

val isLoading: Boolean by viewModel.isLoading.observeAsState(initial = false)

val isCrearContratoSelected: Boolean by viewModel.isCrearContratoSelected.observeAsState(initial = false)

val isContratosPendientesSelected: Boolean by viewModel.isContratosPendientesSelected.observeAsState(initial = false)

val isContratosRegistradosSelected: Boolean by viewModel.isContratosRegistradosSelected.observeAsState(initial = false)

val coroutineScope = rememberCoroutineScope()

if (isLoading) {

Box(Modifier.*fillMaxSize*()) **{**

CircularProgressIndicator(Modifier.*align*(Alignment.Center))

**}**

} else {

Column(modifier = modifier) **{**

HeaderImage(

Modifier

.*align*(Alignment.CenterHorizontally)

.*size*(height = 200.*dp*, width = 200.*dp*)

)

Text(

text = "CONDOSA",

Modifier

.*align*(alignment = Alignment.CenterHorizontally)

.*padding*(bottom = 50.*dp*), fontSize = 32.*sp*, color = Color.White

)

Spacer(modifier = Modifier.*padding*(4.*dp*))

// Botones

Row(

modifier = Modifier

.*fillMaxWidth*()

.*padding*(horizontal = 16.*dp*),

horizontalArrangement = Arrangement.SpaceBetween

) **{**

EmpleadoButton("Crear Contrato") **{**

coroutineScope.*launch* **{**

viewModel.onCrearContratoSelected()

**}**

**}**

EmpleadoButton("Contratos Pendientes") **{**

coroutineScope.*launch* **{**

viewModel.onContratosPendientesSelected()

**}**

**}**

EmpleadoButton("Contratos Registrados") **{**

coroutineScope.*launch* **{**

viewModel.onContratosRegistradosSelected()

**}**

**}**

**}**

Spacer(modifier = Modifier.*padding*(16.*dp*))

//AQUÍIIIIIIIIIII PONES TU TABLAAAAAAAAAAAAAAAAAAAAAAA

**}**

}

}

@Composable

fun EmpleadoButton(text: String, onClick: () -> Unit) {

Button(

onClick = onClick,

modifier = Modifier

.*width*(130.*dp*)

.*height*(60.*dp*),

colors = ButtonDefaults.buttonColors(

contentColor = Color.White,

disabledContentColor = Color.White

)

) **{**

Text(text = text)

**}**

}

@Composable

fun HeaderImage(modifier: Modifier) {

Image(

painter = painterResource(id = R.drawable.*condosa\_1*),

contentDescription = "Header",

modifier = modifier

)

}

**EmpleadoViewModel**

package com.example.mvvmprueba.ui.VistaEmpleado

import androidx.lifecycle.LiveData

import androidx.lifecycle.MutableLiveData

import androidx.lifecycle.ViewModel

import kotlinx.coroutines.delay

class EmpleadoViewModel : ViewModel() {

private val \_isLoading = MutableLiveData<Boolean>()

val isLoading: LiveData<Boolean> = \_isLoading

private val \_isCrearContratoSelected = MutableLiveData<Boolean>()

val isCrearContratoSelected: LiveData<Boolean> = \_isCrearContratoSelected

private val \_isContratosPendientesSelected = MutableLiveData<Boolean>()

val isContratosPendientesSelected: LiveData<Boolean> = \_isContratosPendientesSelected

private val \_isContratosRegistradosSelected = MutableLiveData<Boolean>()

val isContratosRegistradosSelected: LiveData<Boolean> = \_isContratosRegistradosSelected

suspend fun onCrearContratoSelected() {

\_isLoading.*value* = true

delay(1000)

\_isCrearContratoSelected.*value* = true

\_isLoading.*value* = false

}

suspend fun onContratosPendientesSelected() {

\_isLoading.*value* = true

delay(1000)

\_isContratosPendientesSelected.*value* = true

\_isLoading.*value* = false

}

suspend fun onContratosRegistradosSelected() {

\_isLoading.*value* = true

delay(1000)

\_isContratosRegistradosSelected.*value* = true

\_isLoading.*value* = false

}

}

**Carpeta VistaEmpleadoFirma**

**EmpleadoFirmaScreen.kt**

package com.example.mvvmprueba.ui.VistaEmpleadoFirma

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.Button

import androidx.compose.material3.ButtonDefaults

import androidx.compose.material3.CircularProgressIndicator

import androidx.compose.material3.Text

import androidx.compose.runtime.\*

import androidx.compose.runtime.livedata.observeAsState

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import kotlinx.coroutines.launch

import com.example.mvvmprueba.R

import androidx.compose.ui.res.painterResource

import androidx.compose.foundation.Image

@Composable

fun EmpleadoFirmaScreen(viewModel: EmpleadoFirmaViewModel) {

Column(

modifier = Modifier

.*fillMaxSize*()

.*background*(*Color*(0xFF1c4c96))

) **{**

HeaderImage(

Modifier

.*fillMaxWidth*()

.*padding*(top = 16.*dp*, start = 16.*dp*, end = 16.*dp*)

)

RectanguloConTexto()

Tabla()

BotonesSubirFirmaHuella(viewModel)

Spacer(modifier = Modifier.*height*(16.*dp*))

FirmarBoton()

**}**

}

@Composable

fun HeaderImage(modifier: Modifier) {

Image(

painter = painterResource(id = R.drawable.*condosa\_1*),

contentDescription = "Header",

modifier = modifier

)

}

@Composable

fun RectanguloConTexto() {

Box(

modifier = Modifier

.*fillMaxWidth*()

.*background*(*Color*(0xFF6799E3))

.*padding*(20.*dp*)

) **{**

Text(

text = "CONTRATO DE PRESTACIÓN DE SERVICIO",

color = Color.Black,

fontSize = 20.*sp*,

modifier = Modifier.*align*(Alignment.Center)

)

**}**

}

@Composable

fun Tabla() {

// Aquí puedes agregar la implementación de la tabla según tus necesidades

}

@Composable

fun BotonesSubirFirmaHuella(viewModel: EmpleadoFirmaViewModel) {

val coroutineScope = rememberCoroutineScope()

Row(

modifier = Modifier

.*fillMaxWidth*()

.*padding*(16.*dp*),

horizontalArrangement = Arrangement.SpaceBetween

) **{**

Boton("Subir FIRMA", *Color*(0xFF6799E3)) **{**

coroutineScope.*launch* **{**

viewModel.onSubirFirma()

**}**

**}**

Boton("Subir HUELLA", *Color*(0xFF6799E3)) **{**

coroutineScope.*launch* **{**

viewModel.onSubirHuella()

**}**

**}**

**}**

}

@Composable

fun FirmarBoton() {

Button(

onClick = **{ }**,

modifier = Modifier

.*fillMaxWidth*()

.*height*(80.*dp*)

.*padding*(16.*dp*),

colors = ButtonDefaults.buttonColors(

/\*backgroundColor = Color(0xFF1c4c96),\*/

contentColor = Color.White

)

) **{**

Text(text = "FIRMAR", fontSize = 18.*sp*)

**}**

}

@Composable

fun Boton(texto: String, colorFondo: Color, onClick: () -> Unit) {

Button(

onClick = onClick,

modifier = Modifier

.*width*(130.*dp*)

.*height*(60.*dp*),

colors = ButtonDefaults.buttonColors(

/\*backgroundColor = colorFondo, \*/

contentColor = Color.White

)

) **{**

Text(text = texto)

**}**

}

**EmpleadoFirmaViewModel**

**package com.example.mvvmprueba.ui.VistaEmpleadoFirma**

**import androidx.lifecycle.LiveData**

**import androidx.lifecycle.MutableLiveData**

**import androidx.lifecycle.ViewModel**

**import kotlinx.coroutines.delay**

**class EmpleadoFirmaViewModel : ViewModel() {**

**private val \_isLoading = MutableLiveData<Boolean>()**

**val isLoading: LiveData<Boolean> = \_isLoading**

**private val \_isFirmaUploaded = MutableLiveData<Boolean>()**

**val isFirmaUploaded: LiveData<Boolean> = \_isFirmaUploaded**

**private val \_isHuellaUploaded = MutableLiveData<Boolean>()**

**val isHuellaUploaded: LiveData<Boolean> = \_isHuellaUploaded**

**suspend fun onSubirFirma() {**

**\_isLoading.*value* = true**

**delay(1000) // Simulación de carga**

**\_isFirmaUploaded.*value* = true**

**\_isLoading.*value* = false**

**}**

**suspend fun onSubirHuella() {**

**\_isLoading.*value* = true**

**delay(1000) // Simulación de carga**

**\_isHuellaUploaded.*value* = true**

**\_isLoading.*value* = false**

**}**

**}**