



UNIVERSIDAD CENTRAL DEL  
ECUADOR

FACULTAD DE INGENIERIA Y  
CIENCIAS APLICADAS

COMPUTACIÓN

DISPOSITIVOS MOVILES

GRUPO 3

ANDRES CUEVA

MICHAEL CEVALLOS

JHON ENRIQUEZ

MANUEL LLUMIQUINGA

ANDRES LARA

BELÉN PACA

TEMA:  
CONCESIONARIA

QUITO – ECUADOR

2025-2025



## Tabla de Contenido

1. Logo.....	3
2. Tipo de Navegación .....	3
3. Lógica de Navegación (Flujo de Pantallas) .....	3
Pantalla de Inicio de Sesión (PantallaLogin) .....	3
Pantalla de Registro (RegisterScreen).....	5
<b>Editar Vehículos.....</b>	<b>7</b>
<b>Vehículo editado.....</b>	<b>7</b>
<b>Agregar Vehiculo .....</b>	<b>8</b>
<b>Eliminar Vehiculo .....</b>	<b>9</b>
<b>Nuevo Listado.....</b>	<b>9</b>

## Documentación del Modelo de Navegación de la Aplicación DRIVEONIX

### Pantalla de Inicio de Sesión (PantallaLogin)

- **Propósito:** Permite autenticar, ingresando como credenciales su nombre y su apellido
- **Navegación Entrante:** Es la **pantalla inicial** de la aplicación (screen = "login" al inicio). También se accede a ella desde la pantalla de Registro (RegisterScreen) y desde la pantalla de Inicio (PantallaInicio) al cerrar sesión.

### Cifrar contraseña

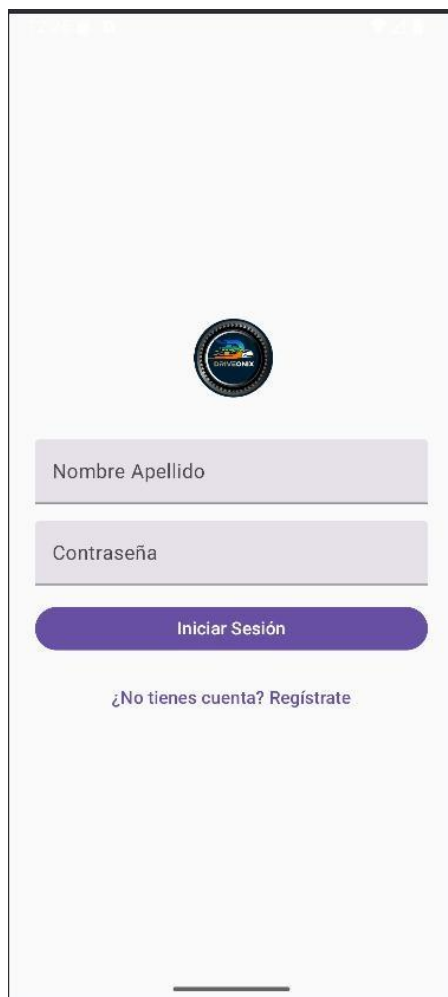
Cómo se cifró la contraseña con SHA-256?

Para proteger la contraseña del usuario, usé un algoritmo llamado SHA-256. Básicamente, este algoritmo sirve para convertir cualquier texto (como la contraseña que escribe el usuario) en una cadena larga de números y letras que no se puede leer ni revertir a su forma original. A eso se le llama hash. Cuando el usuario inicia sesión, guarda su contraseña normal en una variable. Luego, esa contraseña la paso por una función que hice en Kotlin que se llama hashPassword(). Esta función usa una herramienta de Java llamada MessageDigest que viene con el algoritmo SHA-256 incluido.

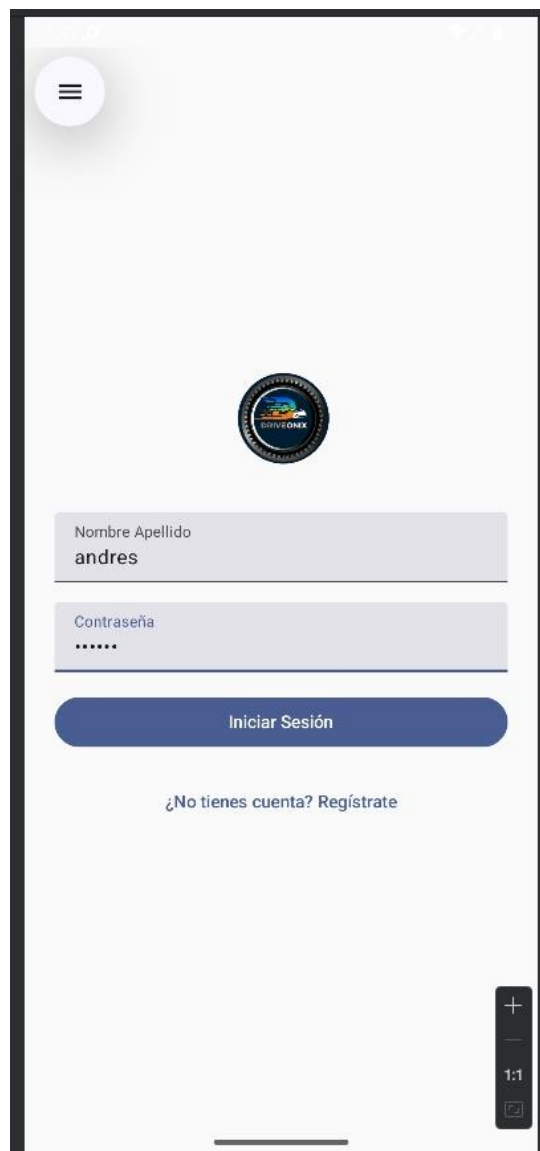
El resultado es una nueva cadena que representa la contraseña cifrada

### Navegación Saliente:

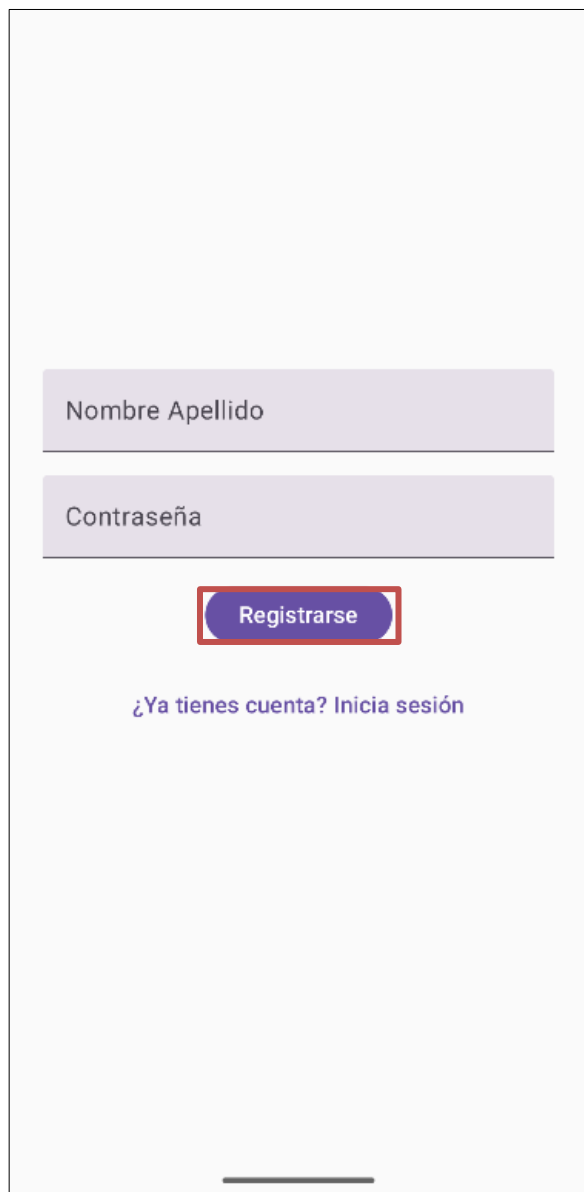
- **onLoginSuccess:** Si el inicio de sesión es exitoso, navega a la **Pantalla de Inicio (PantallaInicio)**. (screen = "inicio")
- **onGoToRegister:** Permite al usuario navegar a la **Pantalla de Registro (RegisterScreen)**. (screen = "register")



Mobile app login screen mockup. The screen features a circular logo at the top center. Below the logo are two input fields: "Nombre Apellido" and "Contraseña". A purple button labeled "Iniciar Sesión" is positioned below the input fields. At the bottom, there is a link that reads "¿No tienes cuenta? Regístrate".



Mobile app login screen mockup with user input. The screen features a circular logo at the top center. Below the logo are two input fields: "Nombre Apellido" (containing the text "andres") and "Contraseña" (containing six dots). A blue button labeled "Iniciar Sesión" is positioned below the input fields. At the bottom, there is a link that reads "¿No tienes cuenta? Regístrate". A hamburger menu icon is visible in the top left corner, and a vertical toolbar with a plus sign, minus sign, and a 1:1 icon is visible in the bottom right corner.



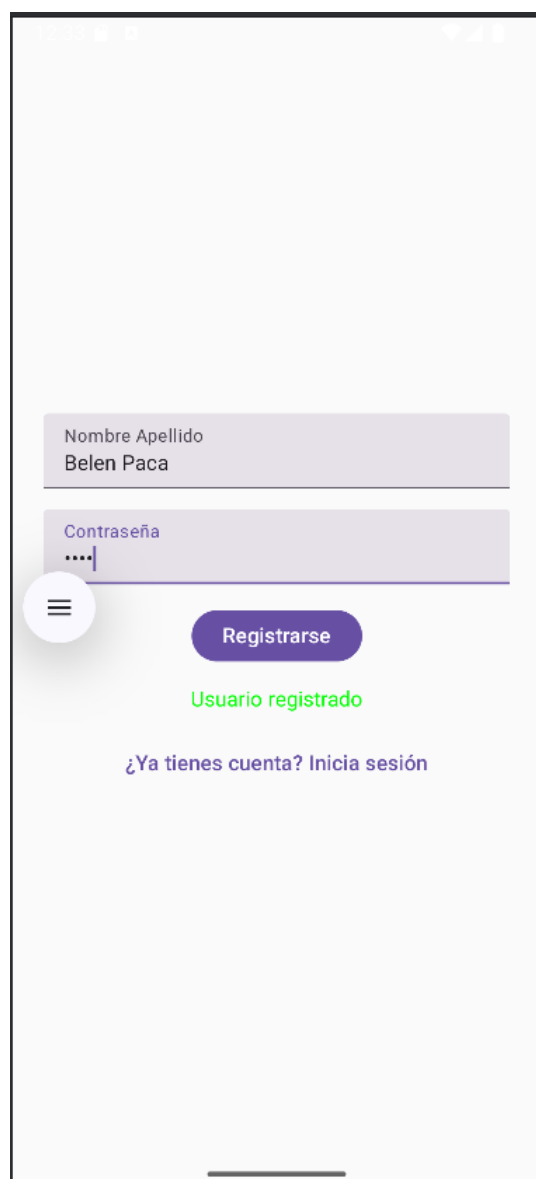
Mobile app registration screen. It features two light purple input fields: "Nombre Apellido" and "Contraseña". Below the fields is a purple "Registrarse" button with a red border. At the bottom, there is a link that says "¿Ya tienes cuenta? Inicia sesión".

Nombre Apellido

Contraseña

Registrarse

¿Ya tienes cuenta? Inicia sesión



Mobile app registration screen showing the state after registration. The "Nombre Apellido" field now contains the text "Belen Paca". The "Contraseña" field contains three dots, indicating it is hidden. A purple "Registrarse" button is present. A green message "Usuario registrado" is displayed above the login link "¿Ya tienes cuenta? Inicia sesión". A white circular menu icon with three horizontal lines is positioned to the left of the "Registrarse" button. The top status bar shows the time as 10:10 and battery level at 80%.

Nombre Apellido  
Belen Paca

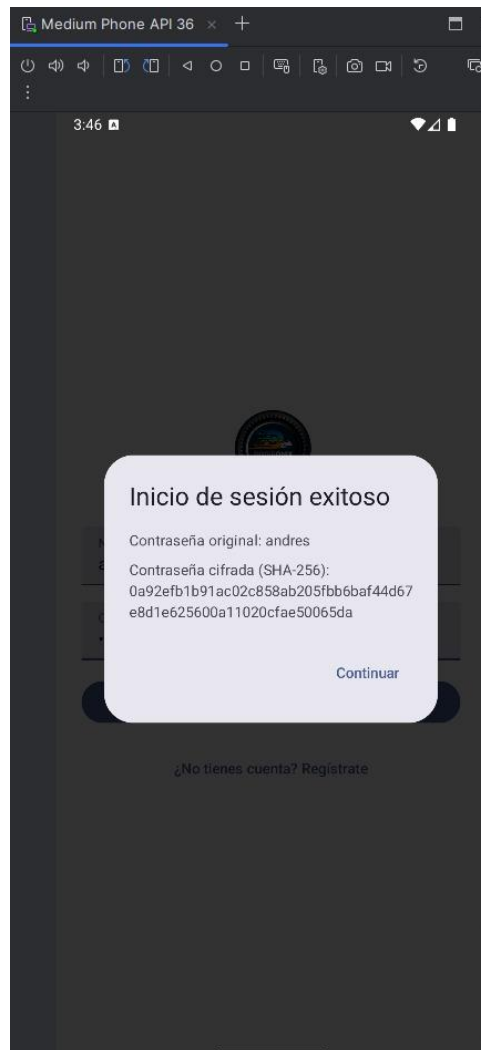
Contraseña  
...

Registrarse

Usuario registrado

¿Ya tienes cuenta? Inicia sesión

## Contraseña Cifrada



Método para ci

```
fun hashPassword(password: String): String {  
    val bytes = MessageDigest.getInstance("SHA-256").digest(password.toByteArray())  
    return bytes.joinToString("") { "%02x".format(it) }  
}
```

